Nicholas M. Patrikalakis
George A. Kriezis

Piecewise Continuous Algebraic Surfaces in Terms of B-Splines

MIT Sea Grant
College Program

Massachusetts
Institute of Technology
Cambridge, MA 02139

MITSG 88-5
August, 1988

PIECEWISE CONTINUOUS ALGEBRAIC SURFACES IN TERMS OF B-SPLINES

by

Nicholas M. Patrikalakis

and

George A. Kriezis

MIT Sea Grant
College Program

Massachusetts Institute
of Technology
77 Massachusetts Ave.
Cambridge, MA 02139

MITSG 88-5
August, 1988
NA86AA-D-SG089
RT-16

Table of Contents

List of Figures

List of Tables

Abstract

In this report we describe a method for representing shape using portions of algebraic surfaces bounded by rectangular boxes described in terms of triple product Bernstein polynomials and we outline some of their properties. The method is extended to handle piecewise continuous algebraic surfaces within rectangular boxes defined in terms of triple products of B-spline basis functions. Next two techniques for sculptured shape creation using primitive data are studied. The first is based on geometric manipulation of existing primitives and the second on approximation/interpolation of lower dimensional entities using least squares techniques. In addition, several interrogation techniques used in the creation and analysis of piecewise continuous algebraic surfaces are presented, such as contouring, ray tracing, volume and moment of inertia and curvature evaluations.

Acknowledgments

This research on shape representation was supported in part by the MIT Sea Grant College Program and the Doherty Foundation. Tracing of algebraics was supported in part by the MIT Sea Grant College Program, the General Electric Company and the National Science Foundation under grant numbers DMC-8706592 and DMC-8720720.

P. Alourdas, C. Bliek, R. H. Carlson and H. N. Gursoy assisted in the implementation of the algebraic surface editor; B. A. Moran in ray tracing algebraic surfaces and Dr. P. V. Prakash in contouring algebraic surfaces.

Authors

Dr. Nicholas M. Patrikalakis is Assistant Professor of Ocean Engineering at M.I.T., Doherty Professor of Ocean Utilization (1988-1990) and a member of the M.I.T. Ocean Engineering Design Laboratory.

Mr. George A. Kriezis is a doctoral graduate student in the Department of Ocean Engineering at M.I.T. and a member of the Design Laboratory.

RELATED SEA GRANT REPORTS

Computation of Algebraic and Polynomial Parametric Surface Intersections, by N. M. Patrikalakis and P. V. Prakash, MIT Sea Grant Report No 87-19, Cambridge MA, 1987.

1. Introduction and Outline

Over the past two decades, there has been significant work in the area of Computer Aided Design on shape representation using piecewise continuous parametric polynomials. As a result of this research workable methods for representing the shape of objects with sculptured surfaces have evolved. In particular, parametric B-splines and their extension to non-uniform rational forms proved to be the most versatile representations with highly desirable properties, allowing the designer to manipulate dynamically the shape or add detail to refine the shape.

Implicit polynomial (algebraic) representation of shape is an alternative to parametric representation. Algebraic representations of curves and surfaces allow representation of the result of intersection, blending and offset operations within the same class of functions without mathematical approximation[1] [2] [3]. Implicit polynomials also allow the mathematical representation of intersections and offsets of polynomial parametric curves and surfaces[4]. This can be seen from the fact that polynomial parametric curves and surfaces can be converted to algebraic representations[5]. It has been pointed out, however, in the literature [6] [3] that it is impractical to perform this conversion for some important cases such as, for example, the biquadratic and bicubic parametric patch cases. This is due to the high degree of the resulting implicit polynomials and the numerical round-off error in the associated computations such as determinant expansions. For example, using elimination techniques, the intersection of two bicubic parametric patches can, in theory, be represented as an implicit equation of degree 54 in each of the parametric variables of one patch.

Because of the importance of intersection, blending and offset problems in geometric modeling applications and the promise of implicit polynomials in providing solutions to these problems, a number of investigators have suggested the use of algebraics, especially **low order algebraics**, in computer aided design. Some of the more important issues in making the application of algebraic curves and surfaces in modeling possible is the ability to represent and interrogate portions of these entities in a finite domain directly and the ability to modify them in a geometrically intuitive manner. For many engineering applications, the rectangle and the parallelepiped can be employed in practice as bounding boxes for shapes represented by algebraics because they naturally allow the creation of piecewise continuous geometries in terms of spline functions. A capability to construct piecewise continuous shapes is needed in the representation of all but the simplest artifacts. Control and interrogation of such algebraic curve and surface geometries, although complex retains a number of geometrically intuitive features. This idea has been suggested for study by a number of investigators[7 1 8]. The development of this idea involves a change of basis from the monomial to the tensor product bivariate and trivariate Bernstein basis for curves and surfaces within a rectangle or a rectangular box, respectively. Piecewise continuous algebraic curves and surfaces can also be naturally represented by extending the Bernstein to the B-spline basis.

This idea for the case of curves has recently been explored as a means of shape creation[9] and shape interrogation through intersection computation[10 11] by the authors' Laboratory. This reformulation, (i.e. definition within a window together with the basis change) not only naturally permits piecewise continuous representation but also provides geometrical significance to the underlying coefficients, because it allows us to view algebraic curves as intersections of parametric B-spline surfaces and a plane. This interpretation allows us to naturally build on the well established theory of modeling with parametric polynomials and to employ many of the geometric algorithms developed over the last decade in that area. Algebraic B-spline curves offer some of the advantages of parametric B-spline curves allowing intuitive manipulation of geometric shape, local control and addition of detail[9].

Encouraged from the useful properties of algebraic B-spline curves, the application of the above theory to modeling with low order algebraic B-spline patches bounded by rectangular boxes was investigated and is reported in this work. Summarizing the main motivations behind the development of a method of sculptured shape representation using low order algebraic curves and surfaces in the B-spline form are

- the degree reduction in the resulting representation, facilitating computationally complex interrogation problems such as intersections and blends.
- the capability for piecewise continuous shape representation, allowing for continuity or discontinuity control of curves and surfaces useful in the representation of all but the simplest artifacts.
- the geometrical significance of the coefficients in this representation, facilitating intuitive understanding and interactive refinement of the representation.

This work is structured as follows.

Chapter 2 describes a method of representing a finite portion of an algebraic surface within a rectangular box using Bernstein polynomials. Some of the properties of this representation useful in geometric modeling applications are also discussed.

Chapter 3 introduces a generalization to piecewise continuous algebraic surfaces defined in terms of B-splines and outlines their properties and possible applications.

Chapter 4 and 5 develop various techniques for shape creation using our formulation for piecewise continuous algebraic surfaces. Chapter 4 deals with the shape creation problem using modification of existing primitive shapes. This is achieved by weights modification and knot refinement allowing change of shape and addition of detail. Chapter 5 develops direct and iterative least squares techniques to approximate lower dimensional entities with the piecewise algebraic surfaces.

Chapter 6 describes some interrogation techniques used in the creation and analysis of the piecewise continuous algebraic surfaces.

Finally, Chapter 7 summarizes the results of this work and indicates possible applications of the proposed method.

2. Description of an Algebraic Surface Patch within a Rectangular Box

2.1 Representation

In this section a portion of an algebraic surface f(x,y,z)=0 within a rectangular box $R_3 = [a_1,a_2] \times [b_1,b_2] \times [c_1,c_2]$ is analyzed. This analysis is based on earlier work on algebraics, such as a method of surface intersection computations[7 10 11] and a method on algebraic curve and surface representation within triangles and tetrahedra[1 8]. Our method is an extension of piecewise continuous algebraic curve representations within rectangles in terms of B-splines[9].

For algebraic surfaces the implicit function f(x,y,z) is a polynomial in the x, y and z variables, the Cartesian coordinates of a point. Restricting the range of definition of f(x,y,z)=0 to a rectangular box in the x-y-z space allows reformulation of the normally monomial representation of f(x,y,z) to a trivariate Bernstein basis which provides geometrical significance to the resulting coefficients. One of the motivations of this transformation is similar to Bezier's reformulation of Ferguson's parametric polynomial tensor product patches. In this manner, the algebraic surface is expressed as:

$$f_{MNQ}(x,y,z) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{Q} w_{ijk}\, B_{i,M}[u(x)]\, B_{j,N}[v(y)]\, B_{k,Q}[w(z)] = 0 \qquad (1)$$

where

$$u(x) = \frac{x-a_1}{a_2-a_1} \qquad v(y) = \frac{y-b_1}{b_2-b_1} \qquad w(z) = \frac{z-c_1}{c_2-c_1} \qquad (2)$$

and $B_{i,M}$, $B_{j,N}$ and $B_{k,Q}$ are the Bernstein-Bezier polynomials of order M, N and Q, respectively, defined by

$$B_{k,P}(t) = \frac{(P-1)!}{(k-1)!(P-k)!}\, t^{k-1}\, (1-t)^{P-k}, \quad k = 1,2,\ldots P \qquad (3)$$

where $0 \le t \le 1$. The numbers M-1, N-1 and Q-1 are the maximum degrees in each of the variables x, y and z, respectively. The scalars w_{ijk} are the weights of the algebraic surface and, as we will see, have a similar function to the control points in parametric polynomial formulations.

As stated earlier, the above formulation is an extension of a method of description of an algebraic curve f(x,y)=0 portion within a rectangle $R_2 = [a_1,a_2] \times [b_1,b_2]$ as presented in[9]

$$f_{MN}(x,y) = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} B_{i,M}[u(x)] B_{j,N}[v(y)] = 0 \tag{4}$$

where everything is defined as above.

Algebraic surface patches, expressed as above, can be easily visualized, for example, using planar contours perpendicular to each of the coordinate axes x, y or z. Each of these contours is an algebraic curve defined in a rectangle which can be traced using the techniques developed to compute intersections of algebraic and rational polynomial patches, expressed as planar algebraic curves in the parametric domain of the patch[9 10 11].

Let us demonstrate this process by an example, using z contours to display a lower order algebraic surface defined with equation (1). If we want to obtain the surface contour with $z = z_n$, where $c_1 \le z_n \le c_2$, using equation (1) we have

$$f_{MNQ}(x,y,z_n) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{Q} w_{ijk} B_{i,M}[u(x)] B_{j,N}[v(y)] B_{k,Q}[w(z_n)] = 0 \tag{5}$$

which can be rewritten as

$$f_{MNQ}(x,y,z_n) = \sum_{i=1}^{M} \sum_{j=1}^{N} w^{\times}{}_{ij} B_{i,M}[u(x)] B_{j,N}[v(y)] = 0 \tag{6}$$

with

$$w^{\times}{}_{ij} = \sum_{k=1}^{Q} w_{ijk} B_{k,Q}[w(z_n)] \tag{7}$$

the new weight coefficients. Equation (6) is identical to equation (4) and represents an algebraic curve defined in a rectangle. According to the tracing techniques developed in [10 11], it is convenient to visualize equation (4) as the intersection of the explicit surface patch $z = f_{MN}(x,y)$, where $(x,y) \in R_2$, with the plane $z = 0$. The above explicit surface patch can now be recast in the following equivalent parametric tensor product Bezier patch form:

$$\mathbf{R}(u,v) = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{R}_{ij} B_{i,M}(u) B_{j,N}(v) \tag{8}$$

where

$$\mathbf{R}_{ij} = [\, x_i', y_j', w_{ij} \,] \tag{9}$$

$$x_i' = a_1 + \frac{(a_2-a_1)(i-1)}{M-1} \tag{10}$$

$$y_j' = b_1 + \frac{(b_2-b_1)(j-1)}{N-1} \tag{11}$$

and i = 1,2,...M; j = 1,2,...N; and the properties

$$\sum_{k=1}^{P} B_{k,P}(t) = 1 \ \ and \ \ \sum_{k=1}^{P} \frac{k-1}{P-1} B_{k,P}(t) = t \tag{12}$$

have been used.

An alternative way to think of the surface represented by equation (1), which provides geometrical significance to the weights w_{ijk} is similar to the above interpretation of an algebraic curve. Equation (1) can be visualized as the intersection of the explicit hyperpatch (volume) $w = f_{MNQ}(x,y,z)$, where $(x,y,z) \in R_3$, with the hyperplane $w = 0$. The above explicit hyperpatch can now be recast in the following equivalent parametric triple product Bezier form:

$$\mathbf{R}(u,v,w) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{Q} \mathbf{R}_{ijk} B_{i,M}(u) B_{j,N}(v) B_{k,Q}(w) \tag{13}$$

where

$$\mathbf{R}_{ijk} = [\, x_i', y_j', z_k', w_{ijk} \,] \tag{14}$$

a four-dimensional vector,

$$x_i' = a_1 + \frac{(a_2-a_1)(i-1)}{M-1} \tag{15}$$

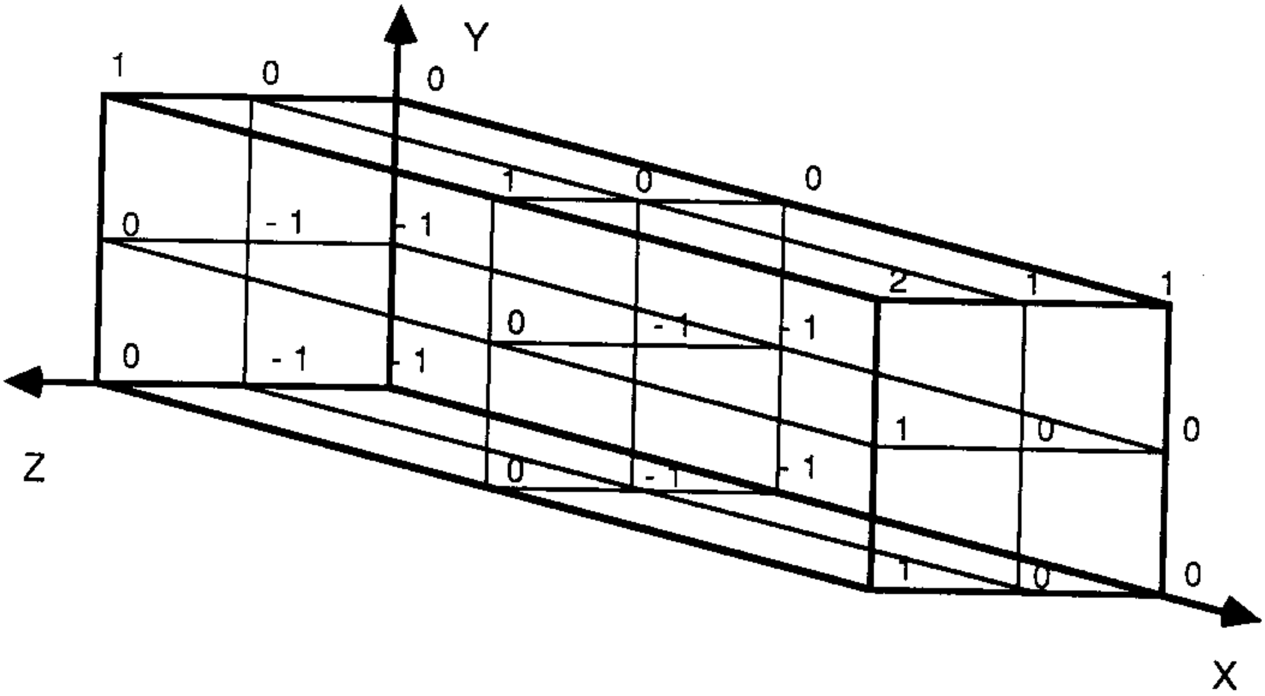
$$y_j' = b_1 + \frac{(b_2-b_1)(j-1)}{N-1} \tag{16}$$

$$z_k' = c_1 + \frac{(c_2-c_1)(k-1)}{Q-1} \tag{17}$$

and $i = 1,2,...M$; $j = 1,2,...N$; $k = 1,2,...Q$; and the properties given in equation (12) were also used.

Equations (13) and (14) now provide a useful interpretation of the coefficients w_{ijk} introduced in equation (1), as the four dimensional coordinate of the control polyhedron vertices of the parametric hyperpatch (volume) given by equation (13). Equations (15) through (17) indicate that the x, y and z coordinates of these vertices are uniformly distributed on a grid defined by $R_3 = [a_1,a_2] \times [b_1,b_2] \times [c_1,c_2]$ with spacings proportional to 1/(M-1), 1/(N-1) and 1/(Q-1) in each of the x, y and z directions and that the projection of $\mathbf{R}(u,v,w)$, $0 \leq u, v, w \leq 1$ on the hyperplane $w = 0$ is identical to R_3. Figure 2-1 illustrates these control vertices (grid points) on a cube with the associated weights for an octant of a sphere.

Figure 2-1: Control Vertices and Weights for an Octant of a Sphere
(Not to Scale)



It is relatively easy to convert an implicit surface to the Bernstein representation. As an example, the portion of the general quadric surface

$$f(x,y,z) = ax^2 + by^2 + cz^2 + 2hxy + 2gzx + 2fyz + 2rx + 2sy + 2tz + d = 0 \qquad (18)$$

within our general rectangular box R_3 involves weights w_{ijk} given by the following general equations :

$$[w_{ij1}] = B\,\Lambda_1(A_1 + c_1A_2 + c_1{}^2A_3)\Lambda_2\,\Gamma$$

$$[w_{ij2}] = B\,\Lambda_1(A_1 + \frac{c_1 + c_2}{2}A_2 + c_1c_2A_3)\Lambda_2\,\Gamma$$

$$[w_{ij3}] = B\,\Lambda_1(A_1 + c_2A_2 + c_2{}^2A_3)\Lambda_2\,\Gamma \qquad (19)$$

where the matrices B, Γ, Λ_1, Λ_2, A_1, A_2 and A_3 are given below

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0.5 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \Gamma = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0.5 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Lambda_1 = \begin{bmatrix} 1 & a_1 & a_1{}^2 \\ 1 & a_2\text{-}a_1 & 2a_1(a_2\text{-}a_1) \\ 0 & 0 & (a_2\text{-}a_1)^2 \end{bmatrix} \qquad \Lambda_2 = \begin{bmatrix} 1 & 1 & 1 \\ b_1 & (b_2\text{-}b_1) & 0 \\ b_1{}^2 & 2b_1(b_2\text{-}b_1) & (b_2\text{-}b_1)^2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} d & 2s & b \\ 2r & 2h & 0 \\ a & 0 & 0 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 2t & 2f & 0 \\ 2g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad A_3 = \begin{bmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

As an example of application of Equation (19), the first octant of a sphere with equation given by $x^2 + y^2 + z^2 - 1 = 0$ involves weights:

$$[\,w_{1jk}\,] = \begin{bmatrix} -1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [\,w_{2jk}\,] = \begin{bmatrix} -1 & -1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [\,w_{3jk}\,] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

also illustrated in Figure 2-1. As a different example, a complete sphere within a unit cube [0,1]x[0,1]x[0,1] given by $(x\text{-}1/2)^2 + (y\text{-}1/2)^2 + (z\text{-}1/2)^2 - 1/4 = 0$ involves weights:

$$[\,w_{1jk}\,] = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & -1/2 & 0 \\ 1/2 & 0 & 1/2 \end{bmatrix} \qquad [\,w_{2jk}\,] = \begin{bmatrix} 0 & -1/2 & 0 \\ -1/2 & -1 & -1/2 \\ 0 & -1/2 & 0 \end{bmatrix} \qquad [\,w_{3jk}\,] = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & -1/2 & 0 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

Figures 2-2 and 2-3 illustrate contours of the two algebraic surfaces defined above.

To generalize equation (19), all algebraic surface portions included within an arbitrary rectangular box can be expressed in the form of equation (1) by a change of basis involving linear combinations of the coefficients of the monomial form of f(x,y,z) = 0. This conversion

Figure 2-2: Contours of the First Octant of a Sphere

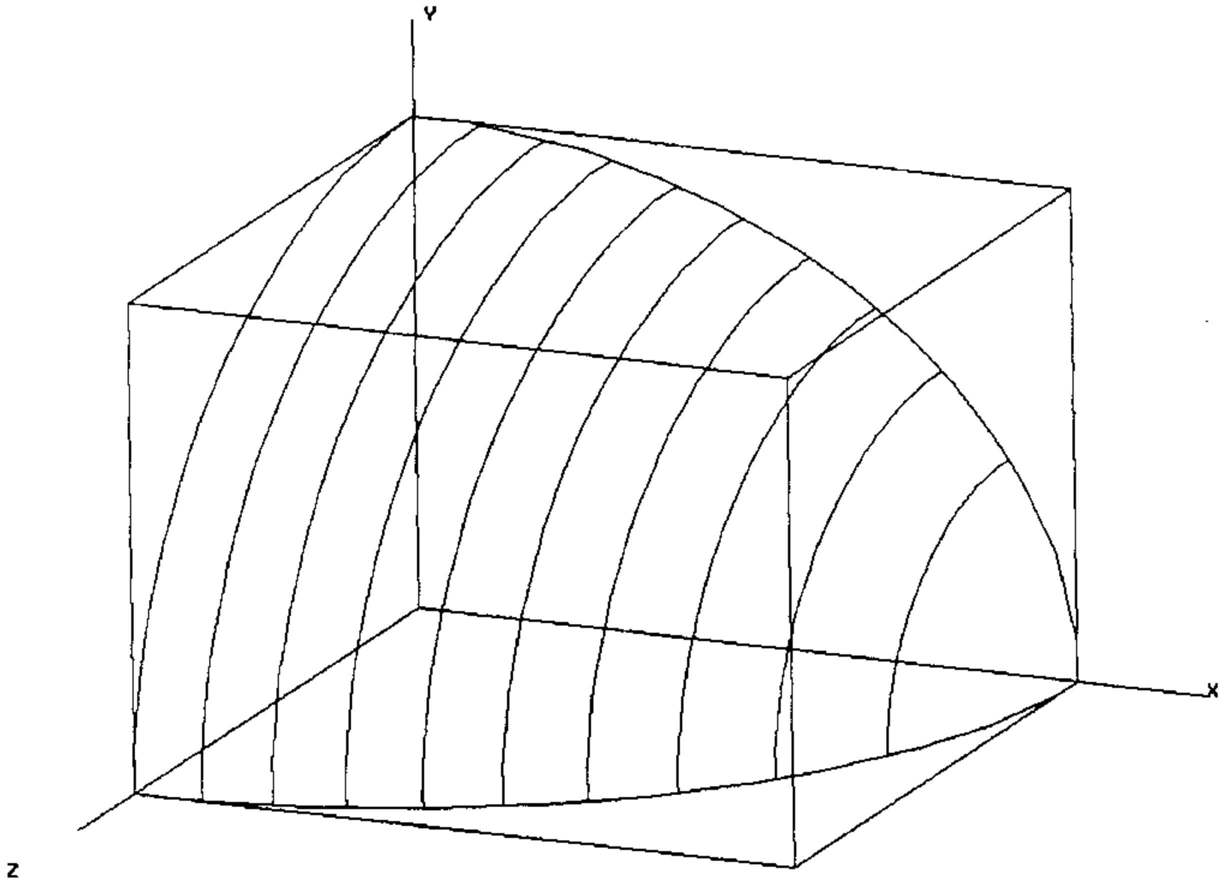
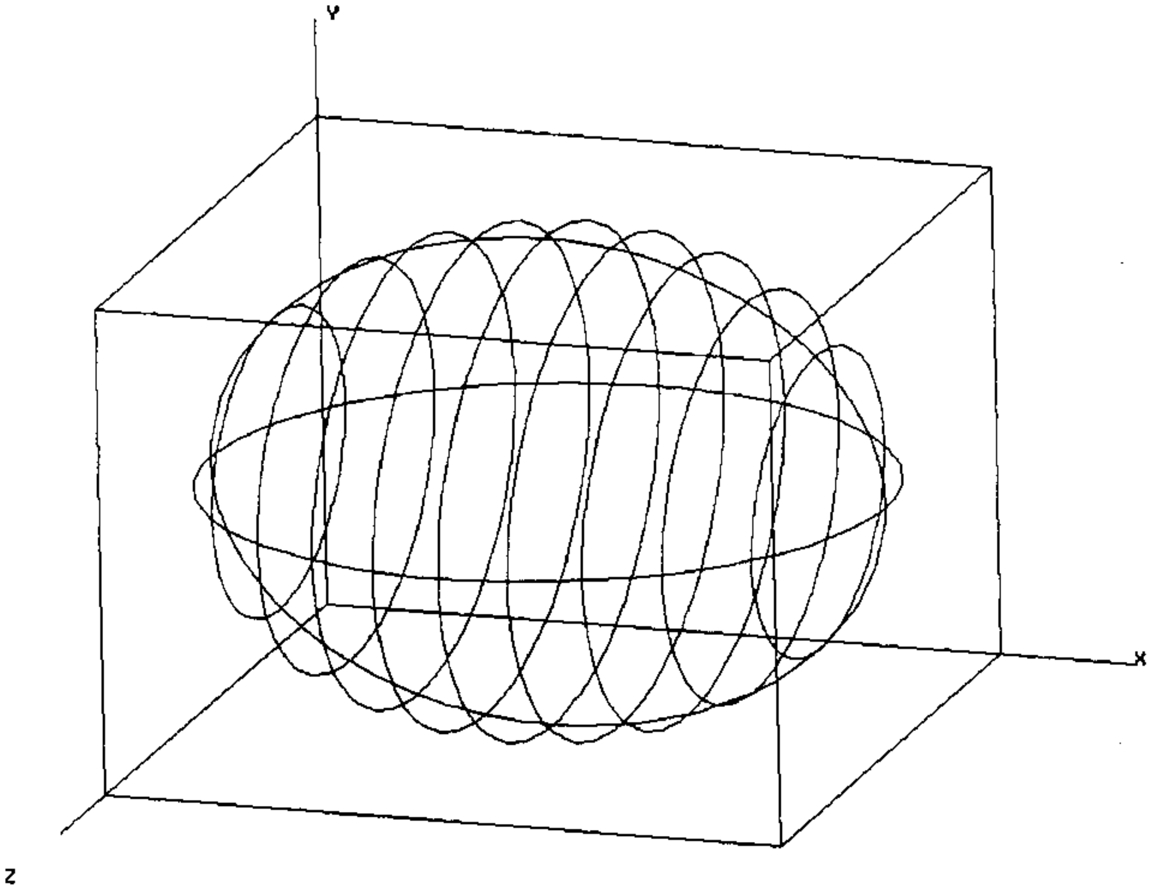


Figure 2-3: Contours of a Sphere



can be reduced to matrix multiplications[3]. The required transformation matrices have elements which are rational numbers depending only on the degree, and, therefore, can be coded in exact arithmetic[10]. Accurate conversion of these rational matrices to floating point matrices is possible by means of a single division for each element. Sophisticated methods to enhance the precision of inner product computation, during basis transformation by matrix multiplication, are available and should be employed[12]. The surface described by equation (1) is of degree M-1 in x, N-1 in y and Q-1 in z, or at most of degree M+N+Q-3 in x, y and z. It is not, however, the most general algebraic surface of degree M+N+Q-3 in x, y and z. It involves only MNQ-1 degrees of freedom because multiples of the weights represent the same surface.

One potential advantage of the Bernstein representation with respect to the power basis representation is increased numerical stability. The condition numbers of simple real roots of a polynomial in an interval have recently been studied theoretically and were found to be smaller in the Bernstein than in the power basis[13]. In addition, Bernstein subdivision and degree elevation procedures decrease root conditioning. The evaluation of Bernstein polynomials using de Casteljau's algorithm has an error bound linear in the degree even though the arithmetic operations grow quadratically with the degree. This error bound is also linear in the maximum of the polynomial coefficients in the Bernstein basis, a notable advantage. Finally, the evaluation of a polynomial is less sensitive to coefficient perturbations in the Bernstein than in the power basis. However, explicit conversion from the power to the Bernstein basis and intermediate polynomial arithmetic in the power basis used in early algorithms in floating point are expected to decrease the benefits inherent to this basis. Practically this means that processing algorithms should adhere to the Bernstein basis [14 10 11].

2.2 Shape Control

Next we present some properties of algebraic surfaces described by equation (1) which should be useful in geometric modeling applications. These properties are similar to the properties of algebraic curves within rectangles[9]. Most of these properties refer to the faces and the edges of the rectangular box and their intersections with the algebraic surface.

If a corner of the box has zero weight, the surface passes through that corner. If the weights corresponding to an edge of the box are all zero, the surface interpolates that edge. If the weights corresponding to a face of the box are all zero, the surface interpolates that face.

If all the weights of the box are strictly positive or negative no portion of the surface exists in R_3. If all weights are strictly positive or negative along a face of the box, the surface has no common point with that face. If all weights are strictly positive or negative along an edge of the box, the surface has no common point with that edge.

If the weights along an edge of the rectangular box have only one sign variation, then the surface intersects that edge at most once. If the weights along a face of the box can be split into two continuous regions with opposite signs, then the surface intersects that face in at most one contour. A simple proof of this fact can be obtained using lines in the face parallel to one of the edges. For each of these lines there will be at most one sign variation in the corresponding weights, and at most one intersection point with the algebraic curve.

Another important property for local control of the surface is the following. If $f(x_i',y_j',z_k')$ is positive (negative), then increasing (decreasing) the corresponding weight w_{ijk} will repel the surface away from (x_i',y_j',z_k') and vice versa decreasing (increasing) w_{ijk} will attract the surface towards (x_i',y_j',z_k'). If all weights on a face of the box increase or decrease monotonically along one of the coordinate directions, then straight lines on the face, parallel to these directions, will intersect the surface at most once. If all weights increase or decrease monotonically along one of the coordinate directions, then straight lines parallel to the corresponding direction intersect the surface at most once. The above properties can also be employed in efficiently tracing algebraic surfaces.

In order to interpolate box corner points with the surface, we have already seen that the corner weight should be zero. The tangent plane to the surface and the normal to the surface at these corner points can then be specified very easily, using the three adjacent weights to the corner weight. The normal vector to an algebraic surface is given by

$$\mathbf{n} = \frac{[f_x, f_y, f_z]}{\sqrt{f_x^2 + f_y^2 + f_z^2}} \tag{20}$$

The normal vector in an interpolated corner point, for example w_{111} has the direction of the vector $\mathbf{n} = [w_{211}, w_{121}, w_{112}]$. If one of these weights is zero, the edge corresponding to that weight is tangent to the surface at the corner point. If two of these weights are zero, the face where the weights lie is tangent to the surface at the corner point. If all these weights are zero, the normal vector is not uniquely defined and we have a singularity of the surface at the corner point.

Piecewise continuous algebraic surface portions within rectangular boxes can be constructed by attaching a new rectangular box to a face of the original box and imposing the necessary continuity conditions on the underlying weights. It is a simple exercise to construct these continuity conditions.

3. Piecewise Continuous Algebraic Surfaces within a Rectangular Box

In this chapter we extend the definition of algebraic surface portions described by equation (1) to more general piecewise continuous algebraic surface portions within a rectangular box $R_3 = [a_1,a_2] \times [b_1,b_2] \times [c_1,c_2]$. We do this using the facilities of the **non-uniform B-spline basis.** Let

$$\begin{aligned} X &= [\, x_1,\, x_2,\ldots\, x_{m+M} \,] \\ Y &= [\, y_1,\, y_2,\ldots\, y_{n+N} \,] \\ Z &= [\, z_1,\, z_2,\ldots\, z_{q+Q} \,] \end{aligned} \tag{21}$$

be three knot vectors involving m+M, n+N and q+Q knots, respectively, chosen so that

$$x_i = \begin{cases} a_1 & i = 1,2\ldots\, M \\ x'_{i-M+1} & i = M+1,\, M+2,\ldots\, m \\ a_2 & i = m+1,m+2,\ldots m+M \end{cases} \tag{22}$$

$$y_j = \begin{cases} b_1 & j = 1,2,\ldots N \\ y'_{j-N+1} & j = N+1,\, N+2,\ldots\, n \\ b_2 & j = n+1,n+2,\ldots\, n+N \end{cases} \tag{23}$$

$$z_k = \begin{cases} c_1 & k = 1,2,\ldots Q \\ z'_{k-Q+1} & k = Q+1,\, Q+2,\ldots\, q \\ c_2 & k = q+1,q+2,\ldots\, q+Q \end{cases} \tag{24}$$

where $x_{M+1} > x_M$, $x_{m+1} > x_m$, $x_{i+1} \ge x_i$, $i = M+1,M+2,\ldots m-1$ and x_i is at most of multiplicity M, and similarly for the other directions. Equation (25) defines a piecewise continuous algebraic surface in terms of B-splines

$$f_{MNQ}^{\,mnq}(x,y,z) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{q} w_{ijk}\, B_{i,M}(x)\, D_{j,N}(y)\, E_{k,Q}(z) = 0 \tag{25}$$

where $B_{i,M}(x)$, $D_{j,N}(y)$ and $E_{k,Q}(z)$ are the non-uniform B-spline bases of order M, N and Q defined on the knot vectors X, Y and Z. These can be evaluated using de Boor's recursion [15] and simplify to equation (1) when m=M, n=N and q=Q so that no interior knots exist[16].

Similarly to equation (1), the surface given by equation (25) can be easily visualized using contouring by planes orthogonal to the coordinate axes. Each of the contours is a piecewise continuous algebraic curve in a rectangle expressed in terms of B-splines. Such curves also arise naturally from intersections of rational parametric B-spline patches with planes. These contours can, therefore, be traced using the techniques presented in[10] and[11].

As in Chapter 2, it is advantageous to view the surface defined by equation (25) as the intersection of the hyperpatch (volume) defined by $w = f_{MNQ}^{mnq}(x,y,z)$ with the hyperplane $w = 0$ where $(x,y,z) \in R_3$. This explicit surface can be recast in a parametric hyperpatch form. To achieve this, we define x_i^*, y_j^* and z_k^* for i=1,2,...m, j=1,2,...n and k=1,2,...q as the expansion coefficients of the functions x, y and z in $[a_1,a_2]$, $[b_1,b_2]$ and $[c_1,c_2]$ respectively in terms of the corresponding B-splines

$$x = \sum_{i=1}^{m} x_i^* \, B_{i,M}(x) \qquad y = \sum_{j=1}^{n} y_j^* \, D_{j,N}(y) \qquad z = \sum_{k=1}^{q} z_k^* \, E_{k,Q}(z) \tag{26}$$

Then these coefficients can be easily obtained using a subdivision algorithm[17]. To do this for the x direction, for example, we define the starting knot vector

$$X^s = [\, a_1, a_1 \ldots a_1, a_2, a_2 \ldots a_2 \,] \tag{27}$$

where the a_1's and a_2's are repeated M times each. The corresponding B-spline basis is now the Bernstein basis of order M (degree M-1) on $[a_1,a_2]$ given by equation (3). Using

$$x = \sum_{i=1}^{M} x_i' \, B_{i,M}(x) \tag{28}$$

which can be obtained from equations (8) to (12), it is now easy to see how starting from the initial representation given by (27) and (28), the final representation given by (21) and (26) is obtained by means of a subdivision algorithm[17].

Alternatively, x_i^*, y_j^* and z_k^* can be computed as the nodes[16] associated with the knot vectors given in equation (21) using

$$x_i^* = \frac{1}{M-1} \sum_{k=i+1}^{i+M-1} x_k \qquad i = 1,2,\ldots m \tag{29}$$

and similarly for the other directions. These formulas degenerate to equations (15), (16) and (17) when m = M, n = N and q = Q. From the above discussion, we can recast the explicit volume $w = f_{MNQ}^{mnq}(x,y,z)$, $(x,y,z) \in R_3$ in the following equivalent parametric tensor product non-uniform B-spline volume patch form:

$$\mathbf{P}(u,v,w) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{q} \mathbf{P}_{ijk} \, B_{i,M}(u) \, D_{j,N}(v) \, E_{k,Q}(w) \tag{30}$$

where

$$\mathbf{P}_{ijk} = [\, x_i^*, y_j^*, z_k^*, w_{ijk} \,] \tag{31}$$

a four-dimensional vector and $(u,v,w) = (x,y,z) \in R_3$. Equations (30) and (31) now provide a useful interpretation of the coefficients w_{ijk} introduced in equation (25) as the w coordinates or the weights of the control polyhedron vertices of the parametric hyperpatch (volume) given in equation (30). Thus the weights may be associated with grid points of the rectangular box which are created using the nodes of the B-spline functions in each of the parametric directions. The geometric interpretation outlined in the previous chapter for the Bernstein basis algebraic surfaces in a rectangular box extends to this case as well.

Examples of algebraic B-spline surfaces are given in Figures 3-1 and 3-2. Figure 3-1 illustrates contours of a piecewise continuous algebraic surface by planes orthogonal to the x axis. The degree of the surface is 2 in each principal direction in the cube [0,1] x [0,1] x [0,1] with a Z knot vector involving a simple interior knot at 0.5, and Y and X knot vectors without interior knots, and weights given by the matrices

$$[\,w_{1jk}\,] = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad [\,w_{2jk}\,] = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad [\,w_{3jk}\,] = \begin{bmatrix} -1 & -1 & -1 & -2 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

The resulting surface involves a ruled segment followed by a curved segment with tangent continuity. The ruled segment is parallel to the z axis and slopes upward in the x direction. The effect of changing weights w_{114}, w_{214} and w_{314} by subtracting the number 4 is illustrated in Figure 3-2. This surface still involves the same ruled segment, but now exhibits a clear inflexion and a much steeper rise from the ruled part, while preserving tangent continuity as before.

As we have seen, piecewise algebraic surfaces given by equation (25), like piecewise parametric surfaces, have the advantages of easy representation of various degrees of continuity, local control, and efficient addition of detail which are inherent in the B-spline formulation[16 18] while retaining the good properties of the Bernstein basis described in the previous section. In fact, due to the generality of equation (25), we have used the B-spline representation in our implementations[10 11].

It is worth noting a few additional properties of equation (25) of interest in geometric modeling applications. The representation given by (25) is invariant under translation, rotation, reflection and uniform scaling because the shape of the surface is related to the shape of the control polyhedron of the hyperpatch and the relative proportions of R_3. These properties are useful in design and graphical visualizations. In addition, because equation

(25) describes a shape enclosed within a rectangular box, it allows for the efficient implementation of interference preprocessing algorithms[19] essential in complex engineering environments. Equation (25) also separates R_3 into regions where $f_{MNQ}{}^{mnq}(x,y,z)$ is either positive or negative, so that it allows the representation of volumes using Boolean combinations of half-spaces in the manner of Constructive Solid Geometry. Finally, this representation of volumes using (25) can be converted naturally to the octree method of representing volumes using subdivision[18] and the convexity properties of B-splines.

Figure 3-1: Piecewise Continuous Algebraic Surface - One Internal Knot in the Z Direction

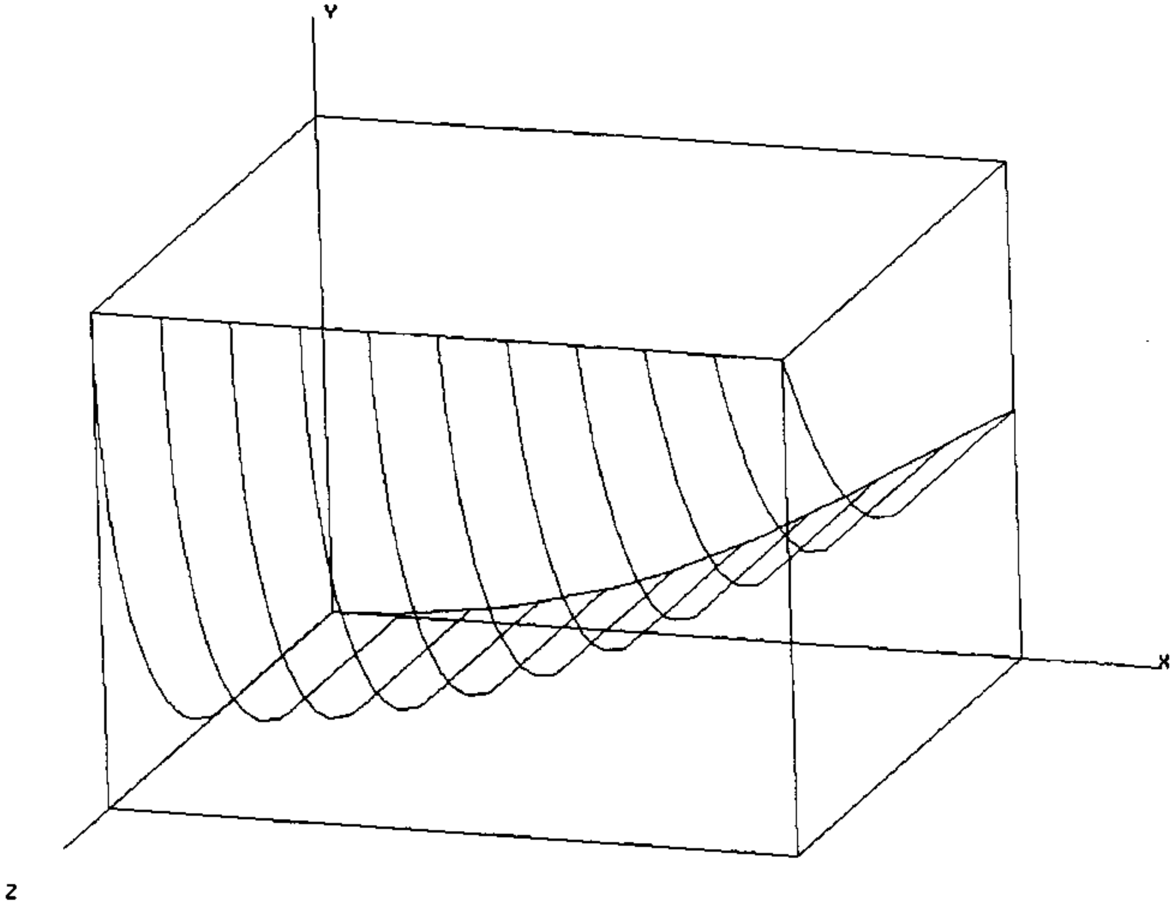
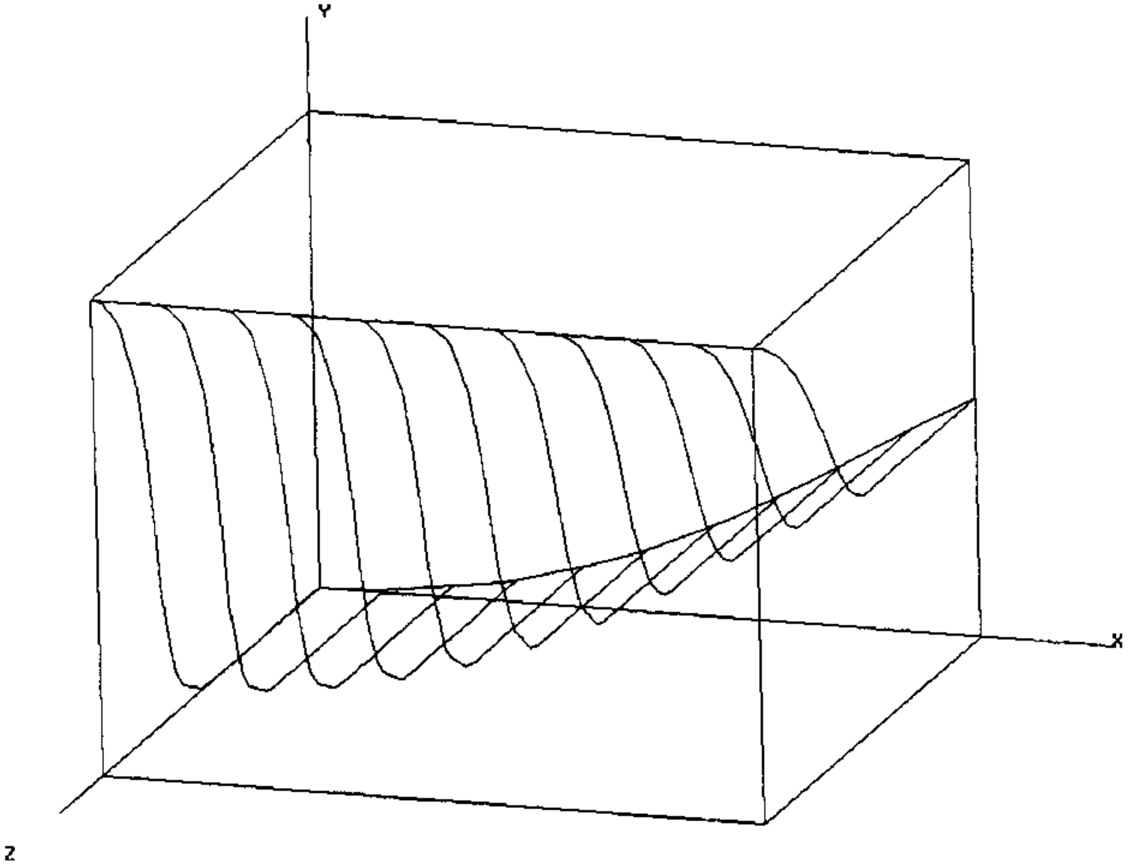


Figure 3-2: Piecewise Continuous Algebraic Surface - One Internal Knot in the Z Direction- Inflection Point



4. Shape Creation I - Manipulation of Primitives

Chapters 4 and 5 concentrate on various techniques to create primitive free-form (sculptured) shape using our piecewise continuous algebraic curve and surface formulations. The actual creation of complex sculptured shapes with algebraic curves and surfaces has received very little attention in the literature[8]. In this work we concentrate primarily on two techniques for shape creation.

- Manipulation of existing primitives
- Approximate least squares fit of lower dimensional entities such as points, control curves etc.

4.1 Manipulation of Existing Primitives

The first idea is to exploit the geometric properties of the B-spline formulation in a rectangular box, such as those presented in Chapters 2 and 3 and for the case of curves in reference[9]. According to this idea we create primitives, starting with pieces of well known surfaces such as quadrics (ellipsoids, cylinders, cones etc.) expressed in the B-spline formulation. This is followed by manipulation of primitive shapes through such operations as non-uniform scaling, knot insertion[17] or degree elevation[20] to increase the degrees of freedom and tweaking of weights to change the shape. In this manner, we may achieve desired shape features and certain geometric constraints (tangencies etc.).

An interactive algebraic surface editor has been developed in our Laboratory for this purpose[21]. In this editor, the designer can start with a well known quadric or with an already designed surface and then apply some of the above operations to increase the degrees of freedom and alter the shape of the surface locally by weight modification, while maintaining continuity in certain portions of the surface.

Some examples of algebraic surface manipulation operations are presented next.

4.1.1 Box Non-Uniform Scaling

Scaling of the rectangular box containing the algebraic surface, scales the surface accordingly, provided the weights are not changed. Thus the sphere given in Figure 2-3 defined inside the box [0,1]x[0,1]x[0,1], becomes an ellipsoid with circular y-z cross-sections inside the box [0,2]x[0,1]x[0,1] (Figure 4-1). An advantage of the present formulation is that a single algebraic Bezier surface in a box can compactly model a complete ellipsoid, or sphere or even a cube (Figure 4-2). As a result, the present algebraic surface formulation in a control box is well suited for use in the framework of Constructive Solid Geometry. Algebraic surfaces in control boxes naturally create finite primitive volumes, which can be procedurally combined in terms of set operators (union, intersection and difference) to create more complex shapes.

4.1.2 Weight Modification

One important technique in modifying shape is changing the weights associated with each of the grid points of the piecewise continuous algebraic surface representation. In Chapter 3 we show an example of the effect of weight modification on a surface with an additional knot in the z direction. In that example (Figures 3-1, 3-2) by changing a few of the weights, we performed some local changes to the surface, we introduced an inflection, while we kept piecewise continuity and a ruled portion on the surface.

An additional example can be seen in Figures 4-3, 4-4 and 4-5. Here, we start with an octant of a sphere, with weights as defined in Chapter 2. First we modify weights w_{133} and w_{233} from 1.0 to 0.114 and weight w_{333} from 2.0 to 1.03 to pull the sphere towards the vertex with coordinates (0,1,1) and edge defined by vertices with coordinates (0,1,1) and (1,1,1). The result is seen in Figure 4-4. Alternatively we modify weights w_{133} and w_{233} from 1.0 to 2.0, weights w_{122} and w_{222} from -1.0 to 1.03 and weight w_{322} from 0.0 to 1.03 to push the sphere away from vertex with coordinates (0,1,1) and edge defined by vertices with coordinates (0,1,1) and (1,1,1). The result is seen in Figure 4-5. In both these operations, we decided not to affect the tangents of the surface in the opposite faces of the control box. The above example illustrates some of the geometric properties of the present formulation as presented in Chapter 2.

Using some of the interrogation techniques developed simultaneously to examine piecewise algebraic surfaces, such as contouring and curvature maps, the designer can interactively evaluate the changes he makes to improve his design.

Figure 4-1: Contours of a Complete Ellipsoid in Box [0,2]x[0,1]x[0,1]

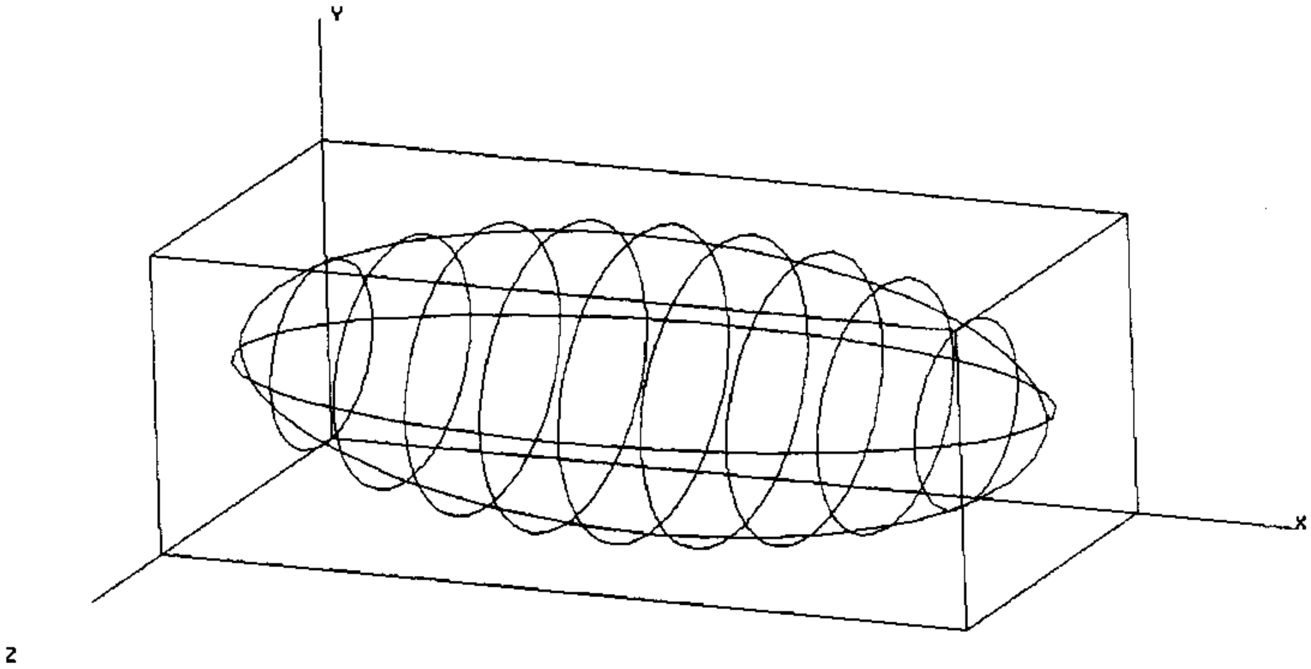


Figure 4-2: Contours of a Cube in Box [0,1]x[0,1]x[0,1]

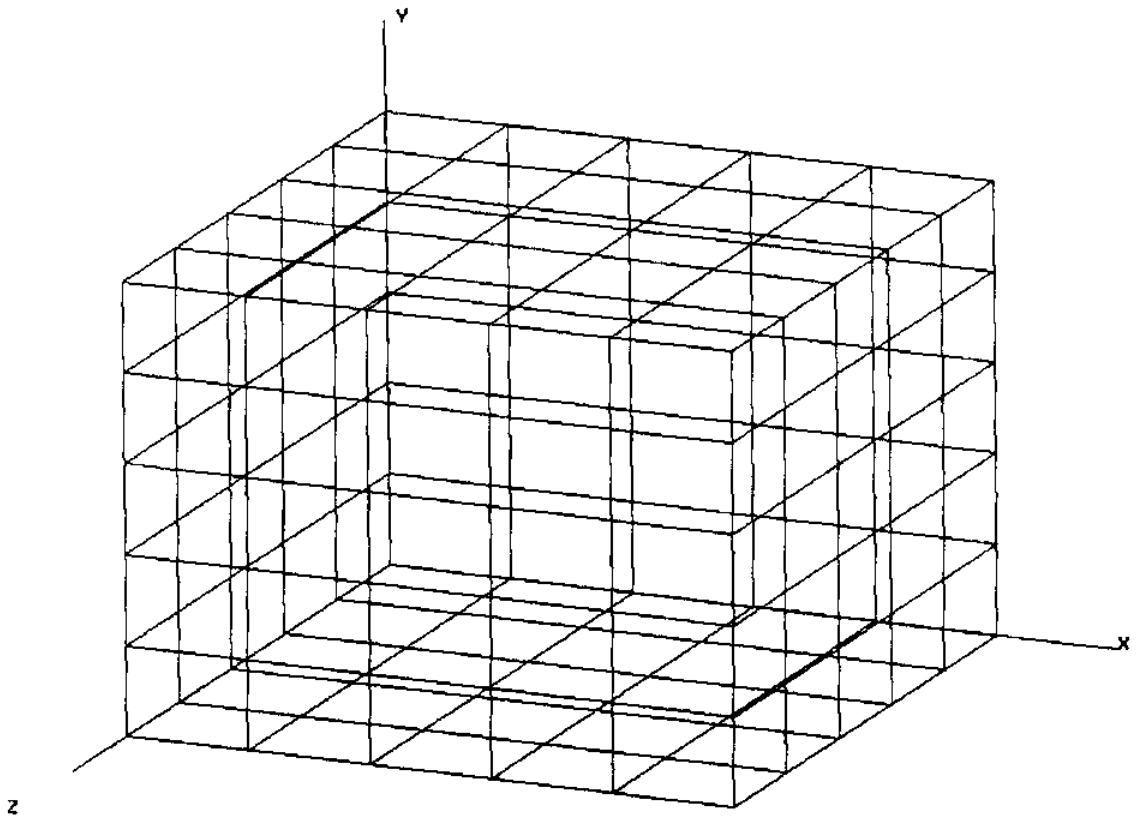


Figure 4-3: Contours of an Octant of a Sphere

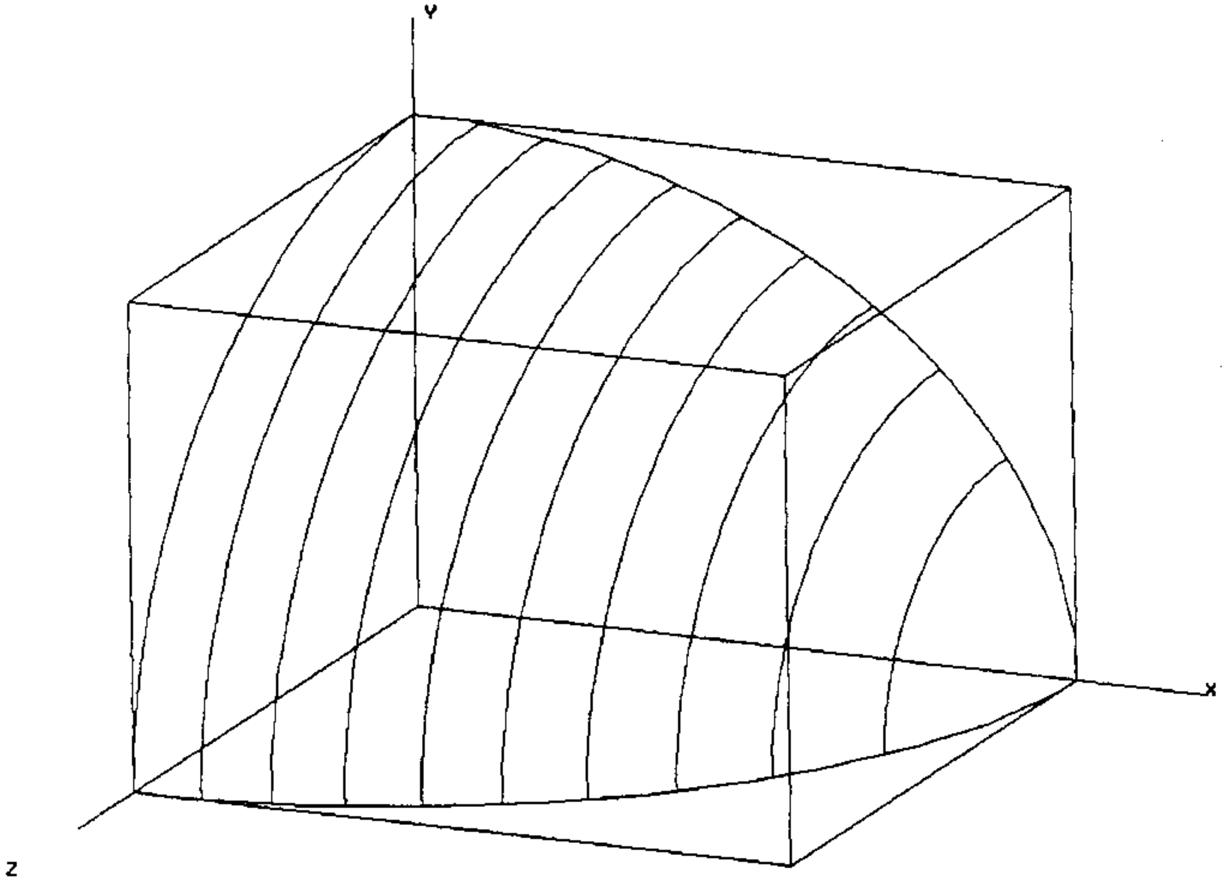


Figure 4-4: Contours of a "Pulled" Octant of a Sphere

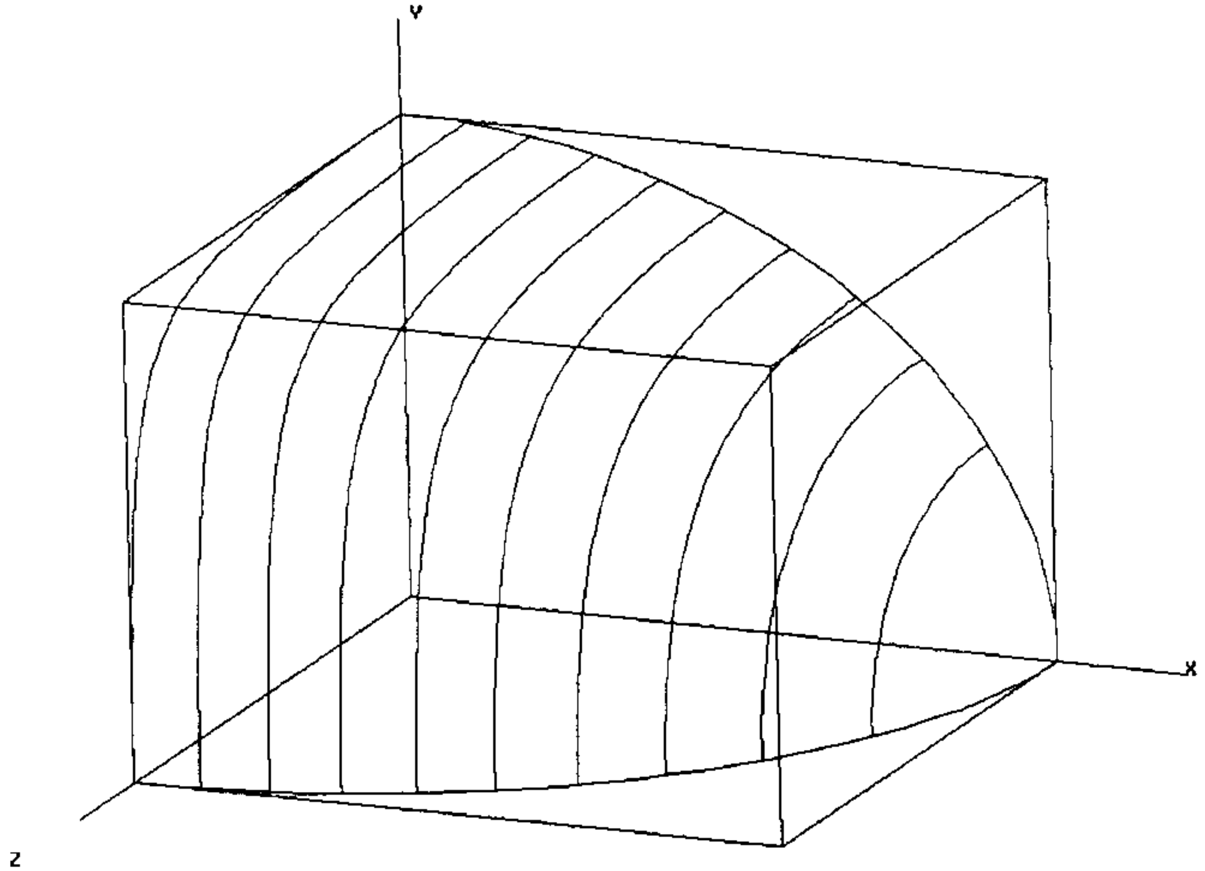
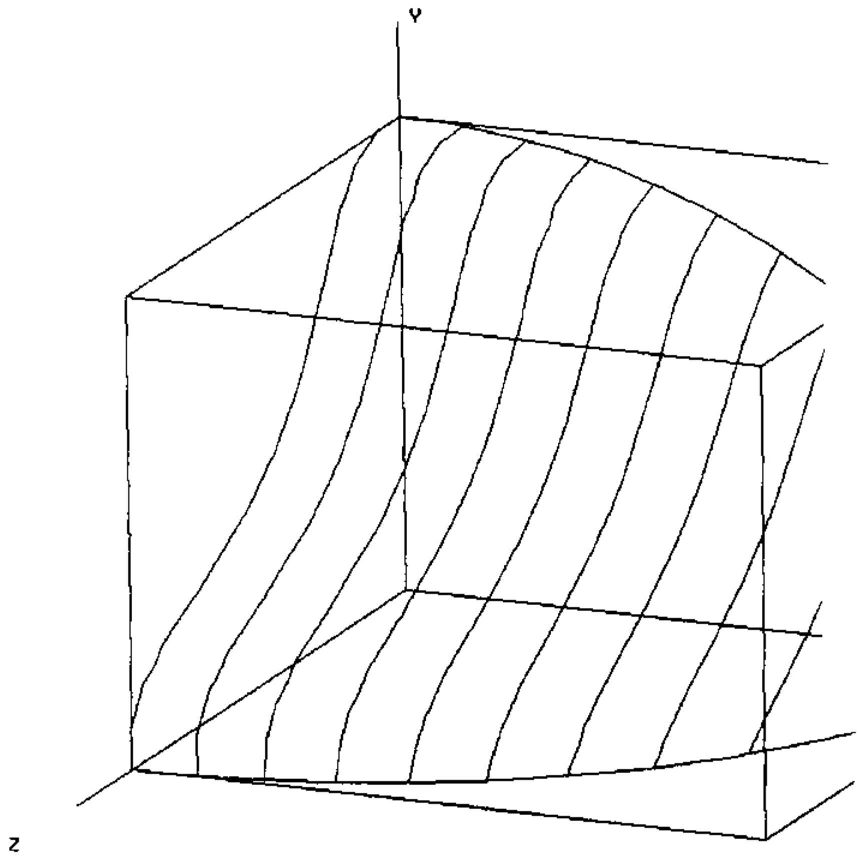


Figure 4-5: Contours of a "Pushed" Oc



4.1.3 Knot Addition and Degree Elevation

An important technique, allowing incremental addition of detail and local shape modification, is the introduction of additional knots in some or all coordinate directions of the control box. This knot addition can be performed using the Oslo Algorithm[18 17] or Boehm's algorithm[22]. Introducing new knots increases the degrees of freedom (weights) of the underlying piecewise surface without actually changing the surface. Modifying the new weights (degrees of freedom), the designer can perform local changes to the surface, while keeping the required degree of continuity between the various segments.

An example of the use of knot addition to locally refine a shape was presented in Chapter 3 and on Figures 3-1 and 3-2, where an initially flat piecewise continuous algebraic surface was used, a knot was added in one of the directions and the weights were modified to affect only a portion of the surface, while the other portion remained ruled.

An additional interactive refinement capability is degree elevation[20] of the algebraic surface, also allowing increase of the degrees of freedom of the surface. However, this technique increases the complexity of the surface and the complexity of the various interrogation and analysis techniques unnecessarily and is not recommended when it leads to degrees beyond 3 in each direction.

The next chapter presents the second method of shape creation, using least squares techniques to approximately fit algebraic curves and surfaces to lower dimensional entities.

5. Shape Creation II - Least Squares Approximation

One idea to create primitive shapes with algebraics involves approximation/interpolation of lower dimensionality entities. For example algebraic B-spline curves may be defined from point data, while algebraic B-spline surfaces may be defined from cross-sectional curves on planes parallel to the control box faces. In order to perform these types of approximations least squares techniques were employed. A description of some of these techniques will be presented next.

5.1 Related Work

There appears to be relatively little written about fitting algebraic curves to points. A fairly thorough search in[23] turned up only a few treatments of least-squares fitting of algebraic curves. Most of these references also treat specific problems (circular, conic fit) and not the general problem which is not as well understood. In addition, only reference [23] was identified as dealing with least-squares fitting of non-planar algebraic surfaces. By comparison least-squares fitting of parametric polynomial curves and surfaces is treated in a number of papers and textbooks, for example[24].

Our work comes to fill part of this significant gap in the literature for the case of fitting of algebraic curves and surfaces. The least-squares technique of[23], a technique based on Householder transformations, as well as a more general technique based on singular value decomposition were implemented to analyze the least squares problem for the case of the piecewise continuous algebraic curves and surfaces of the proposed formulation. The technique based on singular value decomposition was found to be most useful, since it provides more intuition about the least-squares type of solution. We explain this technique in more detail in the following sections. The techniques implemented in this report are of general interest to other scientific domains where curve and surface fitting are needed.

5.2 Problem Formulation

The problem is fitting a general piecewise algebraic curve or surface to m points in two-dimensional or three-dimensional space. The problem formulation will be presented next for algebraic surfaces.

In general, given a sample consisting of m points in R^3, we wish to find a surface of the form $Z(f) \subseteq R^3$ consisting of the zeros of the function $f(x,y,z) = 0$, which comes close to minimize a certain sample-to-surface distance. The function f is to be drawn from a given set Q of functions, which are linearly independent (they form a basis).

The basis functions we use here are the B-spline basis functions. For piecewise algebraic surfaces defined in a rectangular box, the basis functions are the triple products b_l of the B-spline functions in each of the coordinate directions given by

$$b_l(x,y,z) = B_{i,M}(x)\, D_{j,N}(y)\, E_{k,Q}(z) \tag{32}$$

where $B_{i,M}(x)$, $D_{j,N}(y)$ and $E_{k,Q}(z)$, $i=1,...m$, $j=1,...n$, $k=1,...q$, are the non-uniform B-spline bases of order M, N and Q defined on the knot vectors X, Y and Z given in equation (21) and

$$l = i + (j - 1)\, m + (k - 1)\, m\, n$$

with $l = 1,...mnq$. As an example, for a Bezier algebraic curve in a rectangle of degree 2 in each coordinate direction, there are 9 basis functions b_l, while for a Bezier algebraic surface in a rectangle of degree 2 in each coordinate direction, there are 27 basis functions b_l to use in the fit. The addition of knots rapidly increases the number of basis functions.

For piecewise algebraic surfaces, our goal is to determine weights $w_{ijk} = w_l$ such that

$$f(x,y,z) = \sum_{l=1}^{mnq} w_l\, b_l(x,y,z) = 0 \tag{33}$$

comes close to minimize a certain function of the distance of the sample points (x_p,y_p,z_p) from the surface $f(x,y,z) = 0$. To ease notation from now on, we will set $\bar{n} = mnq$ and we will drop the $^-$ as well.

If we are given m points $\mathbf{R_p} = (x_p,y_p,z_p)$ in R^3 to be fitted using the basis functions $b_l(x,y,z)$, we can define the rectangular matrix **A** which will map the basis functions at the m points. Thus **A** is the $m \times n$ matrix ($m > n$, usually $m >> n$), whose ij-th element is

$$a_{ij} = b_j(x_i,y_i,z_i)$$

Equation (33) can then be written in matrix form as

$$\mathbf{A}\, \mathbf{w} = 0 \tag{34}$$

where **w** is the nx1 column vector of the weights.

Problem $\mathbf{A}\mathbf{w} = 0$ is not a standard least squares problem, since the right hand side vector is zero and a trivial exact solution to the problem is $\mathbf{w} = [0]$. This homogeneous problem has received very little attention in the literature, since for standard least squares problems $\mathbf{A}\mathbf{w} =$

b where the objective is to minimize the overall distance vector ||**b** - **Aw**||, the homogeneous part of the solution is always set to zero[24]. In this problem, we want to obtain a non-trivial solution to the homogeneous equation (34), which minimizes the magnitude of the residual vector ||**Aw**||.

We will present three general and formally equivalent methods for solving equation (34). The first method uses singular value decomposition and was found to provide substantial intuition in the solution of equation (34). The second method uses the normal equations of the problem, Cholesky decomposition and determinant evaluations to obtain the solution of equation (34). The second method is faster[23], but less stable and provides little intuition in the solution. A variation of the second method, which will be also presented, uses Householder triangularization of matrix **A** to solve the least squares problem without reverting to the normal equations of matrix **A** and is more stable.

Before moving to the presentation of each of these techniques, the next section describes some distance metrics needed in algebraic least squares problems to evaluate the quality of the fit.

5.3 Distance Metrics

The goodness of fit is usually determined by the length of the m-vector of "distances" of the m points from the approximating surface. There are different definitions of "distance" of a point from an algebraic surface, which are customarily used in least squares problems.

5.3.1 Geometric Distance

Geometric distance of a point p from a surface S is the distance from p to the nearest point of S i.e. the minimum, over all points p' of S, of the Euclidean distance from p to p'. Unfortunately geometric distance is neither computationally nor algebraically convenient. Thus it is customary to use distance metrics approximating geometric distance[23] as identified in Sections 5.3.2 and 5.3.3.

5.3.2 Algebraic Distance

The most common distance metric used for the distance from a point p to a surface Z(f) is the value of f at the point p. Since Z(f) = Z(cf) for $c \neq 0$ ($f(x,y,z) = c \times f(x,y,z) = 0$), f is usually normalized to make this value meaningful, typically by scaling it so as to set to a constant (unity usually) some function of the coefficients (weights). For conics, for example, a quadratic function of the coefficients is usually set to unity (quadratic normalization). Distance computed in this manner is called algebraic distance and is computed by evaluating a fixed representative polynomial cf, chosen independently of the point p. Least squares methods using such a distance metric are called direct methods since they involve no iteration with respect to distance evaluation.

The quality of fit may vary substantially between different normalizations as indicated in[23], since different normalizations introduce different bias to the approximating surface. As an example from conic curves defined by

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

examples of possible normalizations are of the form

$$A^2 + B^2 + C^2 + D^2 + E^2 + F^2 = 1 \quad \text{or}$$
$$F = 1 \quad \text{or}$$
$$A^2 + B^2/2 + C^2 = 2 \quad \text{(for circular fits)}$$

Some of these normalizations have fitting singularities. For example if we try to fit a conic using F=1 to points requiring $F \approx 0$, we will be unable to get a good solution. So conics passing through the origin may not be succesfully fitted with such a normalization.

5.3.3 Non-Algebraic Distance

For the algebraic distance, the normalization technique does not depend on the point $\mathbf{R_p}$. However, in order to approximate geometric distance better, the normalization needs to actually depend on $\mathbf{R_p}$.

In this respect, a first order approximate expression for the shortest distance from any point close to an implicit surface to the surface itself (not in the neighborhood of singularities of f) is

$$|r|_{min} \approx \frac{|f|}{|\nabla f|} \tag{35}$$

where ∇f is the gradient vector of the surface. A proof can be based on a Taylor's expansion of the implicit equation about point $\mathbf{R_p}$, see[11].

Thus in this case, the surface Z(f) can be normalized using $|\nabla f|$ and the distance of the surface from each point can be determined to enable selection of the minimum distance solution. Since this normalization is a function of each of the fitted points $\mathbf{R_p}$, it is called a non-algebraic distance, one though, that is more computationally tractable than geometric distance. This normalization has the beneficial property that it is insensitive to scaling of the surface function f and is as invariant as geometric distance[23].

One disadvantage of this metric, is that it cannot be used directly in a linear least squares fit, since it requires knowledge of the fitted surface and gives rise to a **nonlinear** least squares problem. One computes an algebraic fit using a direct method, and then iteratively weighs the algebraic distance from each sample point $\mathbf{R_p}$ to Z(f) by $\frac{1}{|\nabla f^{(-1)}(p)|}$, where $f^{(-1)}$ is the surface found in the previous iteration. This weighing can be performed by scaling the rows of matrix **A** using the gradient information and then solving the least squares problem with the new **A** matrix. To start the iteration, unit weights can be used or for faster convergence, a direct method can be applied with one of the other distance metrics to obtain a better initial approximation.

5.3.4 Distance Metrics in Present Implementation

In the least squares techniques we studied, we selected to use both algebraic (direct) and non-algebraic distance (iterative).

For the algebraic distance in the direct least squares problem, most of the quadratic normalizations presented in[23] are not applicable to the proposed general least squares fit. In the current implementation two types of normalizations were used. For the implementation of least squares using normal equations or Householder triangularization, the coefficient of one of the basis functions b_l was set to 1 and the other coefficients were scaled accordingly. In this case, there is a singularity, when the corresponding coefficient is actually approximately zero and the results are not expected to be accurate. However, since the coefficients of the algebraic surface in a rectangular box have geometrical significance, it is relatively easy for the designer to select a weight (coefficient) which is non-zero, based on the desired surface or curve geometry. For the implementation of least squares using singular value decomposition, the sum of the squares of the resulting weights (coefficients) of the fit is set equal to 1. This normalization has no singularities.

For the iterative least squares problem (leading to better approximation to the geometric distance between the data points and the fit), the non-algebraic distance was used as defined above, where the algebraic distance from each sample point is weighed using the magnitude of the gradient vectors on the points close to the surface. In addition, the non-algebraic distance was also used as a means to select the more desirable solution vector from a series of feasible solution vectors in the singular value decomposition method as will be explained in the next section. In this case the solution vector minimizing the total non-algebraic distance was selected as the problem solution.

5.4 Least Squares Solution by Singular Value Decomposition

A general technique to solve the least squares problem in equation (34) is singular value decomposition. Applying singular value decomposition to the $m \times n$ matrix **A** we obtain

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^T \tag{36}$$

where **U** and **V** are mxm and nxn orthonormal matrices (i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, the unit matrix and each column u_i of **U** has $||u_i|| = 1$ and similarly for **V**), **S** is an nxn diagonal matrix with positive non-increasing diagonal entries (singular values) starting from the upper left corner, **0** is a (m-n)xn null matrix and $||\ ||$ denotes the Euclidean norm. A proof of the existence of such a transformation, its properties and an algorithm to obtain the transformation can be found in[24]. Some of the most important relevant properties of this decomposition are

- The singular values (s_{ii}) of matrix **A** are the square roots of the eigenvalues of the positive semi-definite nxn symmetric matrix $\mathbf{A}^T\mathbf{A}$ (normal matrix).
- Since matrix $\mathbf{A}^T\mathbf{A}$ and matrix **A** have the same rank, if **A** has rank $k \leq n$, then there are k non-zero eigenvalues of $\mathbf{A}^T\mathbf{A}$ and exactly k non-zero singular values.
- The columns of matrix **V** (right hand singular vectors) are the eigenvectors of matrix $\mathbf{A}^T\mathbf{A}$, while the columns of matrix **U** (left hand singular vectors) are the eigenvectors of matrix $\mathbf{A}\mathbf{A}^T$.
- Let s_1 and s_n, respectively, denote the largest and smallest singular values of mxn matrix **A**, then
  $s_n \leq ||\mathbf{Aw}|| \leq s_1$ provided $||\mathbf{w}|| = 1$
  which is the normalization used in our work. Thus we obtain the lower and upper bound for the residual of all possible fits to the m points.

Another important property, identifying the perturbation properties of singular values, can be drawn from an equivalent property of the perturbation properties of eigenvalues.

> If **A** is a mxn matrix, k is an integer $1 \le k \le m$ and **B** is the (m-1)xn matrix resulting from the deletion of the kth row of **A** (ignoring one of the sample points), then the ordered singular values b_i of **B** interlace with those singular values a_i of **A** as follows
>
> $$a_1 \ge b_1 \ge a_2 \ge b_2 \ge \cdots \ge a_n \ge b_n \ge 0$$

This last property provides an indication of the effect on the singular values of increasing the number m of points in the fit. The singular values do not normally change significantly and their direction of increase or decrease is predictable.

After the singular value decomposition (36), equation (34) can be replaced by the equivalent equation (neglecting the zero rows in the decomposition)

$$\mathbf{S}\,\mathbf{V}^T\,\mathbf{w} = \mathbf{U}^T\,\mathbf{0} = \mathbf{0} \tag{37}$$

which can be written as

$$\mathbf{S}\,\mathbf{p} = \mathbf{0} \tag{38}$$

with

$$\mathbf{p} = \mathbf{V}^T\,\mathbf{w} \tag{39}$$

and

$$\mathbf{w} = \mathbf{V}\,\mathbf{p} \tag{40}$$

Three cases need to be examined for the solution of equations (38) and (40), depending on the rank of matrix **A**.

5.4.1 Rank(A) = n-1

In this case there is an exact fit, since there is a unique non-trivial solution to problem (38). Matrix **S** has exactly n-1 non-zero singular values and $s_n = 0$. The exact solution of (38) (unit vector) in this case is $\mathbf{p}^T = [0, \ldots, 0, 1]$, and the weight vector is $\mathbf{w} = \mathbf{V}\mathbf{p} = v_n$, where v_n is the nth column of matrix **V**, the nth right hand singular vector.

This case is an important case since it provides a direct and, as we also found, accurate method to obtain the implicit equation of a parametric curve or surface, provided of course such an implicit equation exists. An alternate method to compute implicit equations of parametric polynomial curves or surfaces can be found in[5 6].

5.4.2 Rank(A) < n-1

In this case there is still a non-trivial exact fit, but this is not unique. The problem is overdetermined, since the approximating space of functions Q spanned by the basis functions, overdetermines the shape of the input points to be fitted. If rank(A) = r, then **S** has n-r zero singular values and the general solution in this case is

$$\mathbf{p}^T = [0, \ldots, 0_r, a_{r+1}, \ldots, a_n] \textit{ and } \quad \mathbf{w} = \sum_{i=r+1}^{n} a_i \, v_i \tag{41}$$

The weights are a linear combination of the n-r last columns of matrix **V** of right hand singular vectors. The coefficients a_i are not unique, and there is no single solution. This is a difficult case to treat and it is preferable to reduce the order of the approximating implicit polynomial.

5.4.3 Rank(A) = n

In this case (representing the majority of cases) there is no exact fit, since equation (38) has no non-trivial solution and only a fit in the true least squares sense can be obtained. If we select one of the elements of vector **p** to be non-zero (usually unity), the fitting error ρ (magnitude of residual vector) becomes

$$\mathbf{S}\,[0, \ldots, 1_i, \ldots, 0]^T = [0, \ldots, s_i, \ldots, 0]^T$$

and $\rho = s_i$ and the least squares solution vector is $\mathbf{w} = v_i$, the ith column of matrix **V**.

As a result, the singular values are the residual errors if the corresponding right hand singular vectors are used for the fit. If we select the smallest singular value s_n and the associated vector, then the fitting error is minimized and the last right hand singular vector is our least squares solution.

An alternative proof of this observation is the following[24]. If s_j is a singular value of A, then $\mathbf{A}\, v_j = s_j\, u_j$ and since $\|u_j\| = 1$ we get $\|\mathbf{A}\, v_j\| = s_j = \rho$ and s_j is the approximation error when the jth column of **V** is used for the fit weights.

As can be seen in all three cases, the solution of the fitting problem is obtained directly from the singular value decomposition of matrix **A** and particularly from matrix **V**, the matrix of right hand singular vectors.

5.5 Implementation of Singular Value Decomposition Method

Let us present some results from least squares fitting of algebraic curves and surfaces to indicate some of the principles discussed above. First we examine exact fits. If we fit an algebraic curve of order 2 in the x and y direction, to points on a conic curve the fit is exact and the conic curve is recovered. As an example Figure 5-1 presents the fitted curve (8 degrees of freedom) over 10 points which lie on the first quarter of a circle. The fitted curve provides the implicit equation of a circle very accurately. Similarly if we fit an algebraic surface of order 2 in each of the parameter directions to points on a quadric surface the fit is exact and the quadric surface is recovered. As an example Figure 5-2 presents the fitted surface (26 degrees of freedom) over 45 points which lie on the first octant of an ellipsoid. The fitted surface provides the implicit equation of an ellipsoid very accurately.

The existence of the exact fit provides a direct and accurate method to obtain the implicit equation of a parametric curve or surface, provided of course such an implicit equation exists. As an example Figure 5-3 presents the implicit curve (15 degrees of freedom) obtained from fitting of 17 points obtained from a planar fourth order parametric (cubic) Bezier curve with control points (0.0, 0.0), (0.5, 0.0), (0.0, 0.5) and (1.0, 1.0).

The second case examined involves a matrix **A** with rank less than n-1. In this case as was mentioned above there is still a non-trivial exact fit, but this is not unique. The problem is overdetermined, since the approximating space of functions Q spanned by the basis functions, overdetermines the shape of the input points to be fitted. As an example, an algebraic curve of degree 2 in each of the coordinate directions was fitted to points from the parabola $y = 0.9x^2$. Because the degree of y in the parabola is 1, the algebraic curve with degree 2 in the y direction overdetermines the shape of the input curve and the resulting rank of matrix **A** is 7 (n=9). As a result two singular values are zero and there is no unique exact fit. The two right hand singular vector solutions are presented in Figures 5-4 and 5-5 where 21 points were used for the fit. Figure 5-4 presents the expected parabolic solution. Figure 5-5 presents a reducible curve composed of a parabola and a line, i.e. the fit of implicit equation $(y-0.9x^2)(y-0.4264)=0$, which is also a second degree algebraic equation. A linear combination of these two solutions is also a possible exact solution to the fitting problem, i.e. any line parallel to the x axis and the parabola are possible exact fits to the points used for the fit. This example illustrates the difficulty of selecting the best solution from the possible

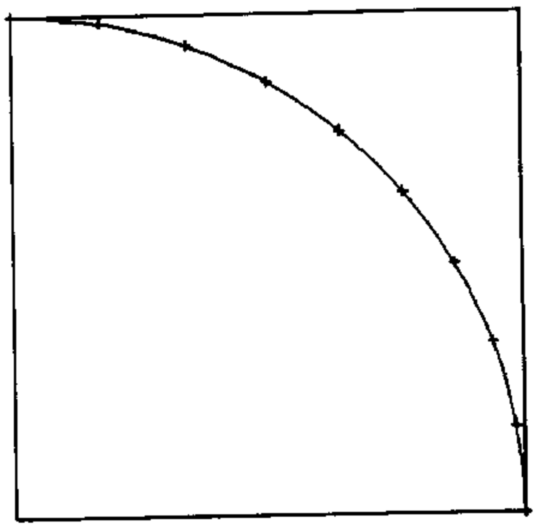
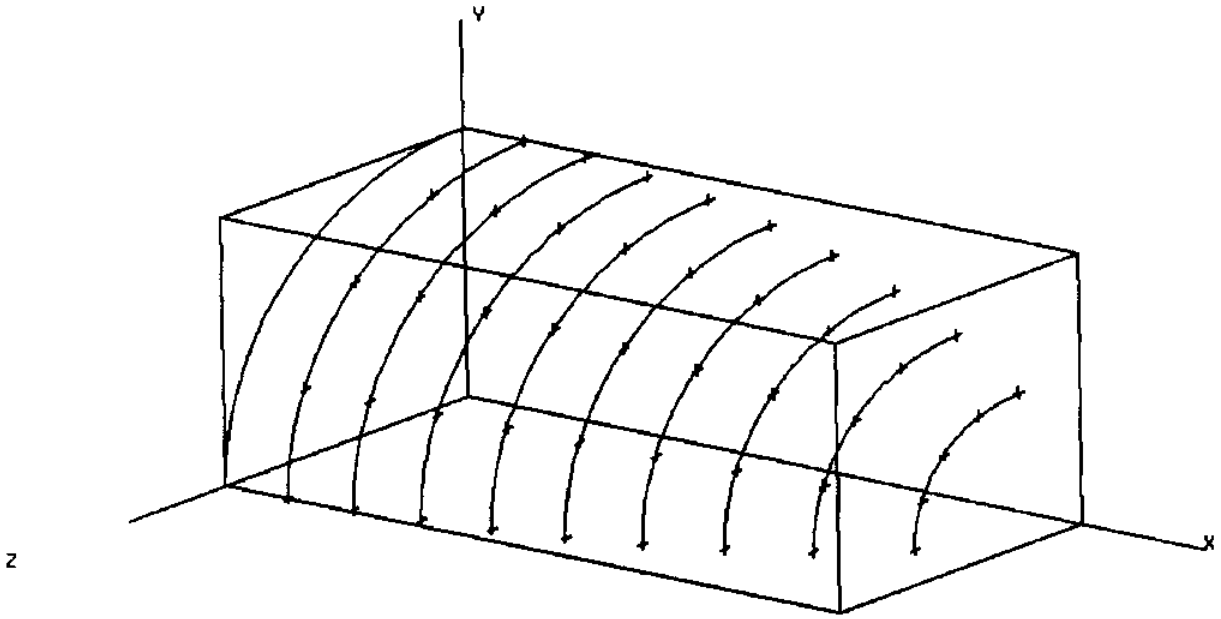


Figure 5-1: Exact Fit of Algebraic Curve to Circle Points

Figure 5-2: Exact Fit of Algebraic Surface to Ellipsoid Points



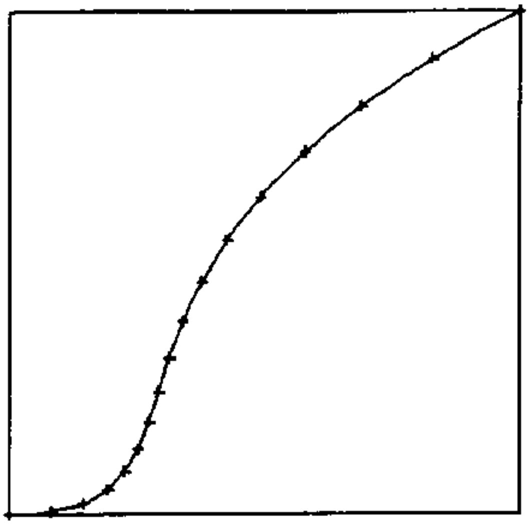


Figure 5-3: Exact Fit of Algebraic Curve to Parametric Bezier Curve Points

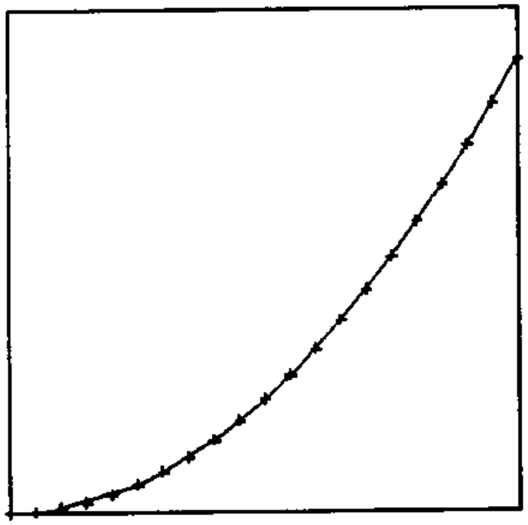


Figure 5-4: First Exact Fit of Algebraic Curve to Parabola Points

exact solutions in an overdetermined fitting problem.

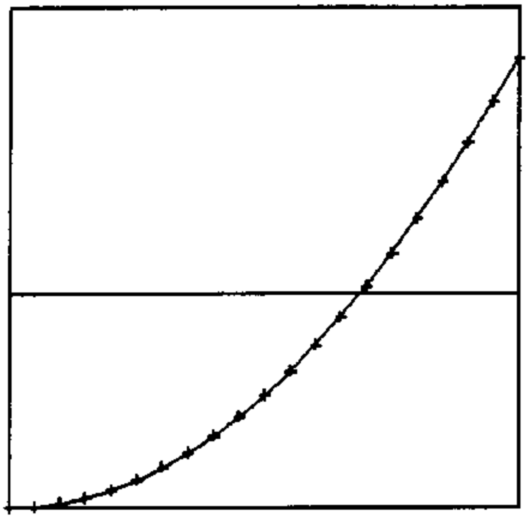


Figure 5-5: Second Exact Fit of Algebraic Curve to Parabola Points

The third and most important case is when the rank of A is n and there is no exact fit but only a fit in the true least squares sense. In this case singular value decomposition provides us with all the necessary information to efficiently determine the best solution. As was already explained above the singular values are the residual errors if the corresponding right hand singular vectors are used for the fit. If we select the smallest singular value s_n and the associated vector, then the fitting error $||Aw||$ is minimized and the last right hand singular vector is our least squares solution.

The main advantage of the singular value decomposition method is that it provides a number of feasible solutions with known residuals and not just the minimum length solution. In the majority of the examples examined a number of singular values are small. In such cases, selecting the smallest singular value for the fit may not be the best solution, since the resulting curve or surface might have undesirable features as explained below.

As an example, if s_n and s_{n-1} are both small and of the same order of magnitude, the 'best' solution might be the vector v_n or the vector v_{n-1} or even a linear combination of v_n and v_{n-1} (a vector in the space spanned by orthonormal vectors v_n and v_{n-1}). If there are more small singular vectors, the problem becomes even more complex since the space of reasonable solutions grows. An additional important consideration is that the singular values are the algebraic distance errors of each of these solutions and do not necessarily approximate accurately the geometric distance between the fitted curve or surface and the points, as will be shown below.

A couple of examples will be presented next to illustrate some of these features. An algebraic Bezier curve of degree 2 in each coordinate direction is used to fit points obtained from a planar third order parametric B-spline curve with knot vector [0,0,0,0.5,1,1,1] and control points (0.0,0.2), (0.4,0.1), (0.7,0.7) and (1.0,1.0). Table 5-1 presents the 9 singular values of the resulting matrix A in decreasing order if we use 21, 31 and 41 points for the fit. The singular vectors corresponding to each singular value in all three cases are very similar. As can be seen, there are several singular values which are small and of the same order of magnitude. This means that several singular vectors approximate the points accurately since their algebraic distance residual is small.

Table 5-1: Singular Values from Fit of Parametric B-Spline Curve

Singular Value	21 points	31 points	41 points
9	0.0019	0.0023	0.0026
8	0.0065	0.0077	0.0087
7	0.0085	0.0101	0.0114
6	0.0501	0.0582	0.0652
5	0.1523	0.1784	0.2010
4	0.4304	0.5057	0.5710
3	0.8566	1.0160	1.1530
2	1.4940	1.7870	2.0370
1	1.7260	2.0730	2.3720

Figures 5-6, 5-7 and 5-8 present the singular vector solutions for the ninth, eighth and seventh singular values respectively. The points used in the fit are also shown in the Figures using crosses. In this case selecting the solution with the smallest singular value, we obtain a solution with two extraneous segments in the unit rectangle which might be undesirable and requires further processing such as subdivision to eliminate extraneous components. The second smallest singular value gives a more disappointing solution with two segments which do not represent the topology correctly. The third smallest singular value gives a solution which has no extraneous segments in the domain considered and approximates the initial points fairly well. It may seem surprising that the eighth solution vector has smaller residual error than the seventh solution vector. However this error is the algebraic distance error (value of the implicit function at the fitting point) and does not approximate geometric distance correctly. Table 5-2 presents the non-algebraic distance error (by dividing the algebraic error of each point by the gradient vector magnitude at the point) for the four smallest error solutions. As can be seen, the eighth solution vector is not as good any more and the seventh solution vector is very close in accuracy to the ninth solution vector.

Table 5-2: Algebraic Distance vs Non-Algebraic Distance Error from Fit of Parametric B-Spline Curve

Singular Vector	Algebraic Distance	Non-Algebraic Distance
9	0.0023	0.0125
8	0.0077	0.0921
7	0.0101	0.0157
6	0.0582	0.1751

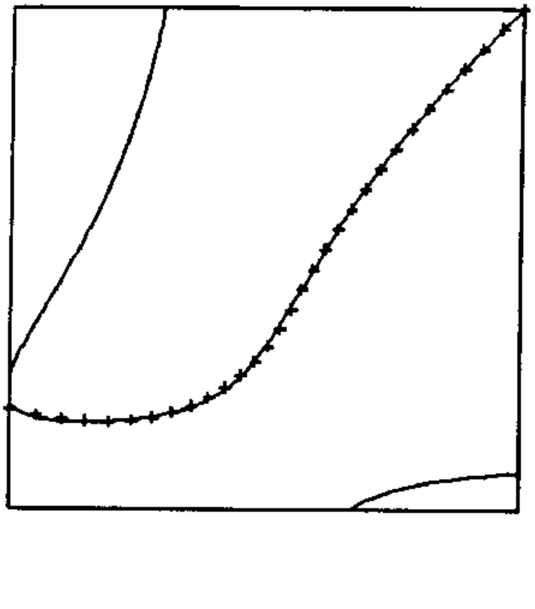


Figure 5-6: Ninth Singular Vector Solution to Fit of B-spline Curve Points Residual Error = 0.0023 (Algebraic Distance)

The second example from algebraic surfaces illustrates some of the features indicated

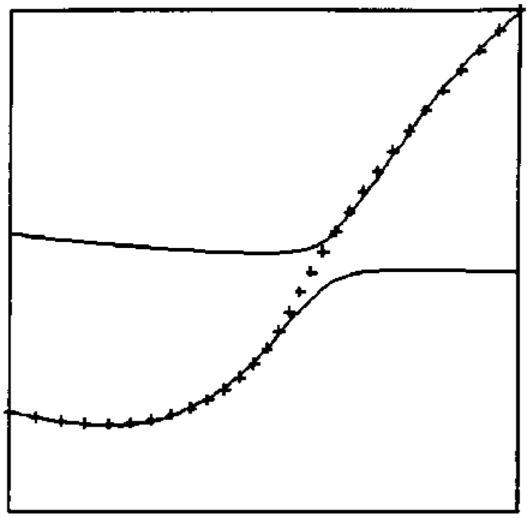


Figure 5-7: Eighth Singular Vector Solution to Fit of B-spline Curve Points
Residual Error = 0.0077 (Algebraic Distance)

above for the case of surfaces. In this example an algebraic Bezier surface of degree two in each of the coordinate directions is used to fit 121 points obtained from a doubly curved integral biquadratic Bezier parametric surface in the box [0,2]x[0,1]x[0,1] with control points P_{1j} = [(0.0 0.0 0.7), (0.0 0.1 0.1), (0.0 0.9 0.0)], P_{2j} = [(1.0 0.0 1.0), (1.0 0.3 0.3), (1.0 1.0 0.7)] and P_{3j} = [(2.0 0.0 0.6), (2.0 0.1 0.1), (2.0 0.9 0.0)]. Matrix **A** in this case has 27 singular values. Table 5-3 presents a comparison of the six smallest and the largest singular values (algebraic distance error) of this problem with the corresponding non-algebraic distance errors. As can be seen in this case also there are a number of solutions with small singular values and, therefore, selection of the smallest singular value for the solution may not be appropriate. As the non-algebraic distance errors indicate, the second smallest singular value solution (26th singular vector) results in the smallest non-algebraic distance error and should therefore be the one selected. Figure 5-9 presents this solution together with the points used in the fit. The solutions corresponding to the 25th and 27th singular vectors contain extraneous surface segments which are undesirable and do not represent the topology correctly. Only the 26th solution vector (Figure 5-9) represents the initial points accurately and at the same time corresponds to the minimum non-algebraic distance error.

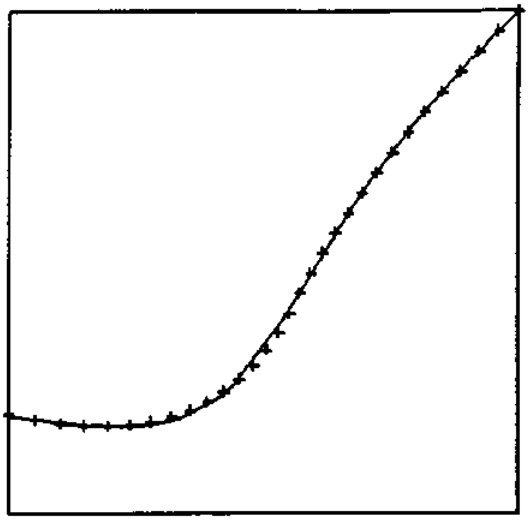
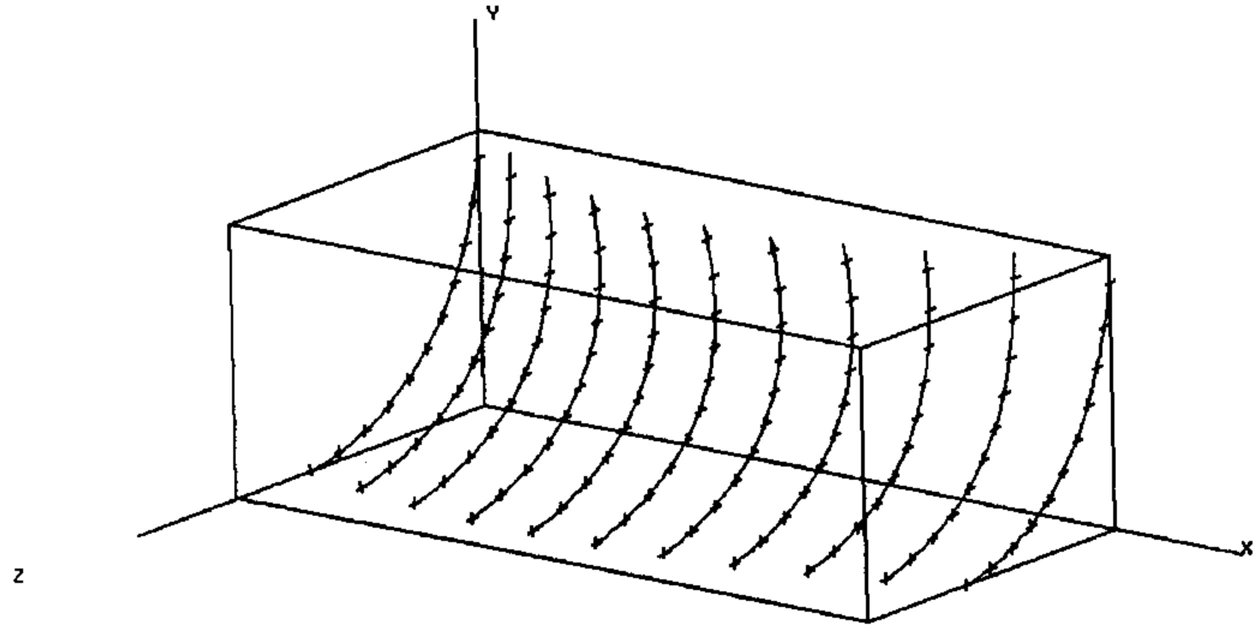


Figure 5-8: Seventh Singular Vector Solution to Fit of B-spline Curve Points
Residual Error = 0.0101 (Algebraic Distance)

Table 5-3: Algebraic Distance vs Non-algebraic Distance Error
from Fit of Parametric Bezier Surface Points

Singular Vector	Algebraic Distance	Non-Algebraic Distance
27	0.00152	0.0836
26	0.00195	0.0052
25	0.00267	0.0922
24	0.00813	0.1969
23	0.01504	N/A
22	0.01824	N/A
1	2.43821	N/A

Figure 5-9: 26th Singular Vector Solution to Fit of Bezier Surface Points
Residual Error = 0.00195 (Algebraic Distance)



As illustrated above, a way to improve on the selection of the correct solution is to use the non-algebraic distance metric or the iterative least squares algorithm, trying to select at each step the solution which minimizes the non-algebraic distance of the sample points from the surface as was described in the previous section. This technique, however, is not able to distinguish extraneous and possibly undesirable curve or surface segments which might exist in the fitted curve or surface in a given domain as the curve example above indicated (Figure 5-6). Such cases require further user interaction and subdivision to eliminate undesirable components.

An alternative method was examined for the selection of the best solution from the space of possible solutions. A linear combination of the singular vectors resulting in small errors was used to interpolate a few sampling points (curve ends etc.) and thereby determine the best solution subject to such constraints. This technique was implemented, but was found not to be very encouraging since it introduces a bias of the solution towards these interpolated points. It is only useful, if the fitted curve or surface is required to pass through specific points. Figures 5-10 and 5-11 present an application of this method. The three smallest error singular vector solutions in the B-spline curve fitting example above were linearly combined to interpolate the two ends of the parametric curve (Figure 5-10) or two center points of the parametric curve (points 13 and 16, Figure 5-11). Figure 5-10 indicates a solution similar to the smaller error solution in Figure 5-6 with the additional segments slightly reduced in size. Figure 5-11 presents a different solution with unacceptable topology.

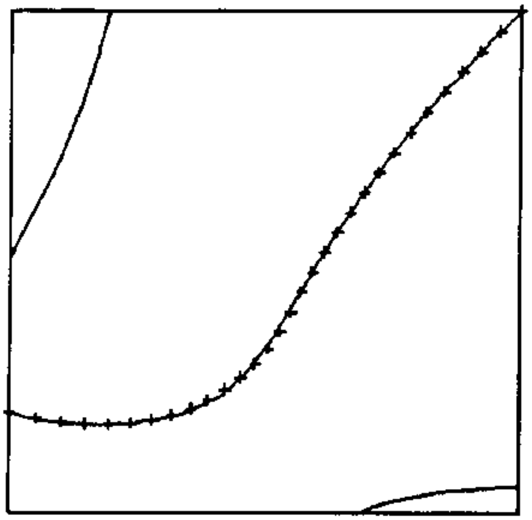


Figure 5-10: Algebraic Curve Solution from Interpolation of Two Ends of B-spline Curve

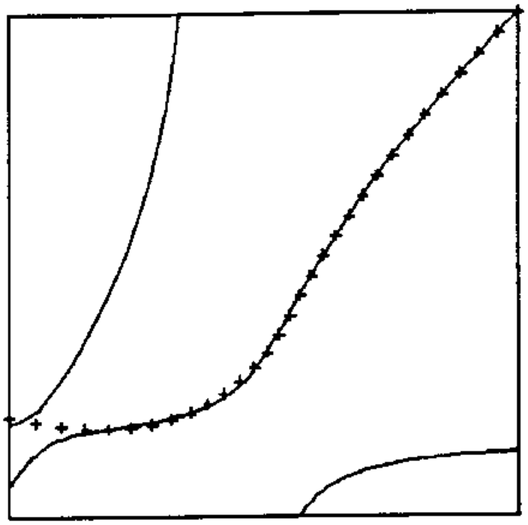


Figure 5-11: Algebraic Curve Solution from Interpolation of Two Interior Points of B-spline Curve

The above examples indicate the complexity of solution of the least squares problem, in which various solutions may need to be examined. Singular value decomposition provides all the necessary information to examine all possible solutions. All the other methods, to be presented next, determine only one solution, usually the one with the smallest residual (corresponding to s_n) and as a result may miss the most appropriate solution. The above examples also indicate the limitation of using the algebraic distance to select the best solution, as compared to the non-algebraic distance which approximates more accurately the geometric distance between the fitted curve or surface and the points.

Some additional and more practical examples will be presented next to indicate the capabilities of representation using piecewise continuous algebraic curves and surfaces. First, piecewise continuous algebraic curves were used to model two-dimensional airfoil sections. Most of the wing sections in common use are either NACA airfoils or have been strongly influenced by the NACA investigations[25]. For our example we used a 4-digit NACA airfoil, specifically NACA 2410 airfoil, with 2% camber at 0.4 of the chord from the leading edge, with section thickness 10% of the chord length[25]. Thirty four points from this section were used for the fit in the rectangle [-0.05,0.95]x[-0.1,0.1]. For simplicity, the trailing edge of the foil was not included in the rectangle as it is normally modelled with straight line segments. A second and a third degree (in each variable) Bezier algebraic curve was used for the fit. The resulting curves are shown in Figure 5-12 for the second degree algebraic curve and in Figure 5-13 for the third degree algebraic curve. The fitted points are also shown in these Figures using crosses. As can be seen a single third degree algebraic curve approximates very well the NACA section, while the same is not true for the second degree curve.

In the second example, piecewise continuous algebraic surfaces were used to approximate a high degree explicit polynomial ship hull form, which is a variation of the simple parabolic hull form first introduced by Wigley. This hull form is given by

$$y(x,z) = ((1 - z^2)(1 - x^2)(1 + 0.2x^2) + 2(1 - z^8)(1 - x^2)^4)/3 \qquad (42)$$

where the coordinates have been non-dimensionalized so that the length coordinate x runs from (-1,1), the depth coordinate z runs from (0,1) and the breadth coordinate runs from 0 on the centerplane to 1 at the maximum breadth on one side. Since equation (42) is linear in y, the approximating surface was selected also linear in y to avoid obtaining a reducible surface

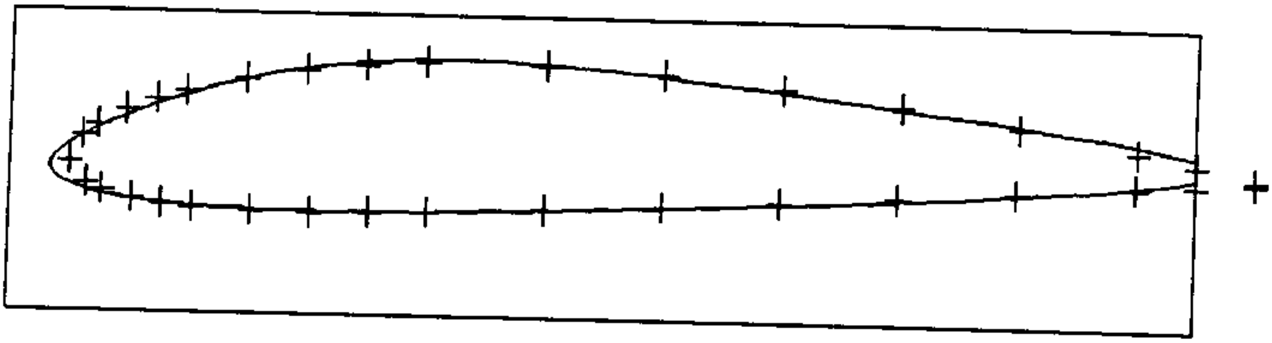


Figure 5-12: Second Degree Algebraic Curve Solution for NACA-2410 Airfoil

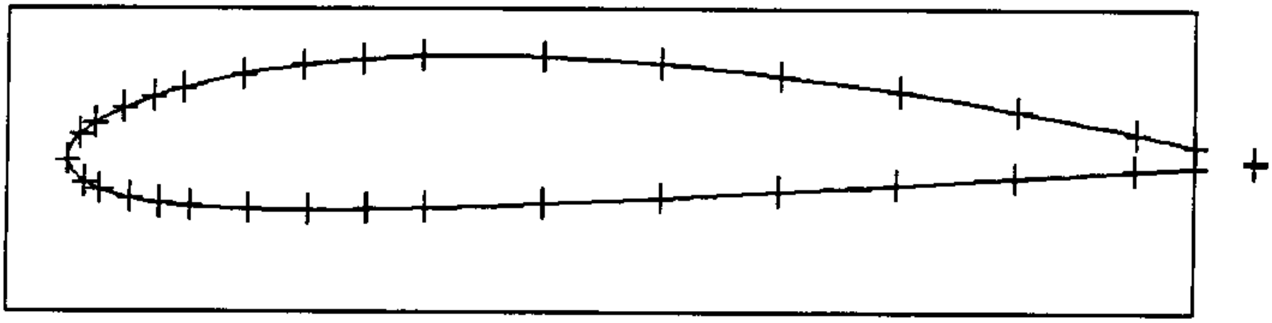


Figure 5-13: Third Degree Algebraic Curve Solution for NACA-2410 Airfoil

solution. A third order in the x and z and a second order in the y algebraic surface was used to approximate 176 points from equation (42) in the control box (0,1)x(0,1)x(0,1) (one quarter of the hull form). A Bezier algebraic surface as well as various piecewise continuous algebraic surfaces were examined. Figures 5-14 and 5-15 present contours for two of the approximating surfaces. Figure 5-14 presents the Bezier solution, while Figure 5-15 presents a B-spline solution with knot vectors in the x and z direction given by [0,0,0,0.5,1,1,1]. Both solutions are very good approximations to the hull form given by equation (42), while the B-spline solution with the larger number of degrees of freedom results in slightly less residual error.

In the third example, an algebraic surface was used to approximate a standard Series 60 ship hull. The ship selected has a length of 400 ft, a half beam of 37.5 ft and a depth of 37.5 ft. The half front section of the ship was approximated using a Bezier algebraic surface defined in the rectangular box [0,200]x[0,40]x[0,40]. One hundred and four points were used (8 points per section, 13 ship sections) for the fit. The small number of points available limited the number of possible degrees of freedom of the approximating surface. A very good solution was obtained using a Bezier algebraic surface of second order in the y direction and third order in the x and z directions (18 degrees of freedom). Figure 5-16 presents this solution, together with the initial points on the ship stations displayed. The approximation is very good, even near the edge of flat of bottom of the Series 60 hull where the approximating surface is trimmed by the plane z=0 but is not precisely tangent to this plane.

These examples indicate some of the capabilities of shape representation using low order algebraic curves and surfaces. In all the examples the selection of each of the solutions from the possible range of solutions was performed using the non-algebraic distance metric described above.

The singular value decomposition technique was implemented using standard numerical routines[26]. The decomposition of matrix **A** is performed in two steps. First matrix **A** is reduced to a triangular matrix **R** using Householder transformations and then the singular value analysis of **R** is performed. The asymptotic number of operations required for this decomposition is $O(mn^2 + 6n^3)$. The singular value computation is equivalent to the complete solution of an eigenproblem and is an iterative computation. As a result the time required for this decomposition is larger than the time required by the other methods to be

Figure 5-14: Bernstein Algebraic Surface Fit of Modified Wigley Hull Form

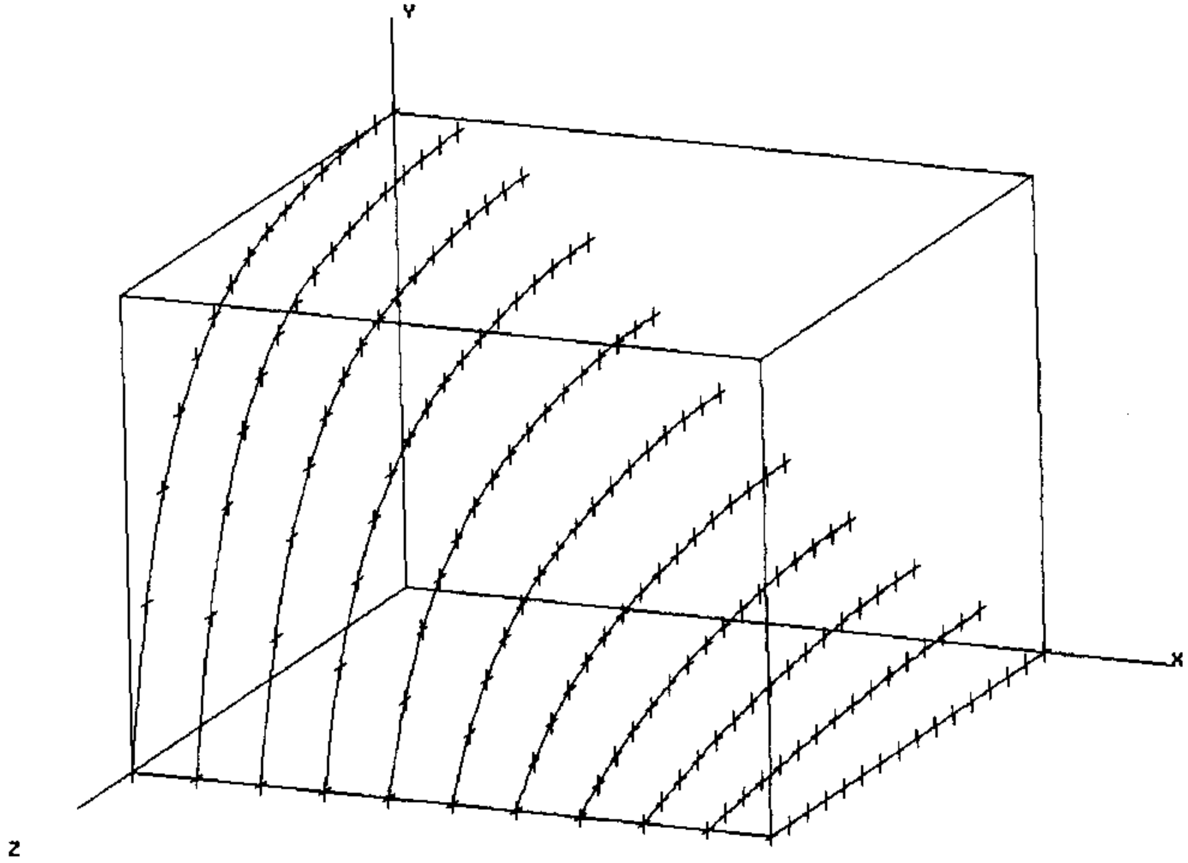


Figure 5-15: B-Spline Algebraic Surface Fit of Modified Wigley Hull Form

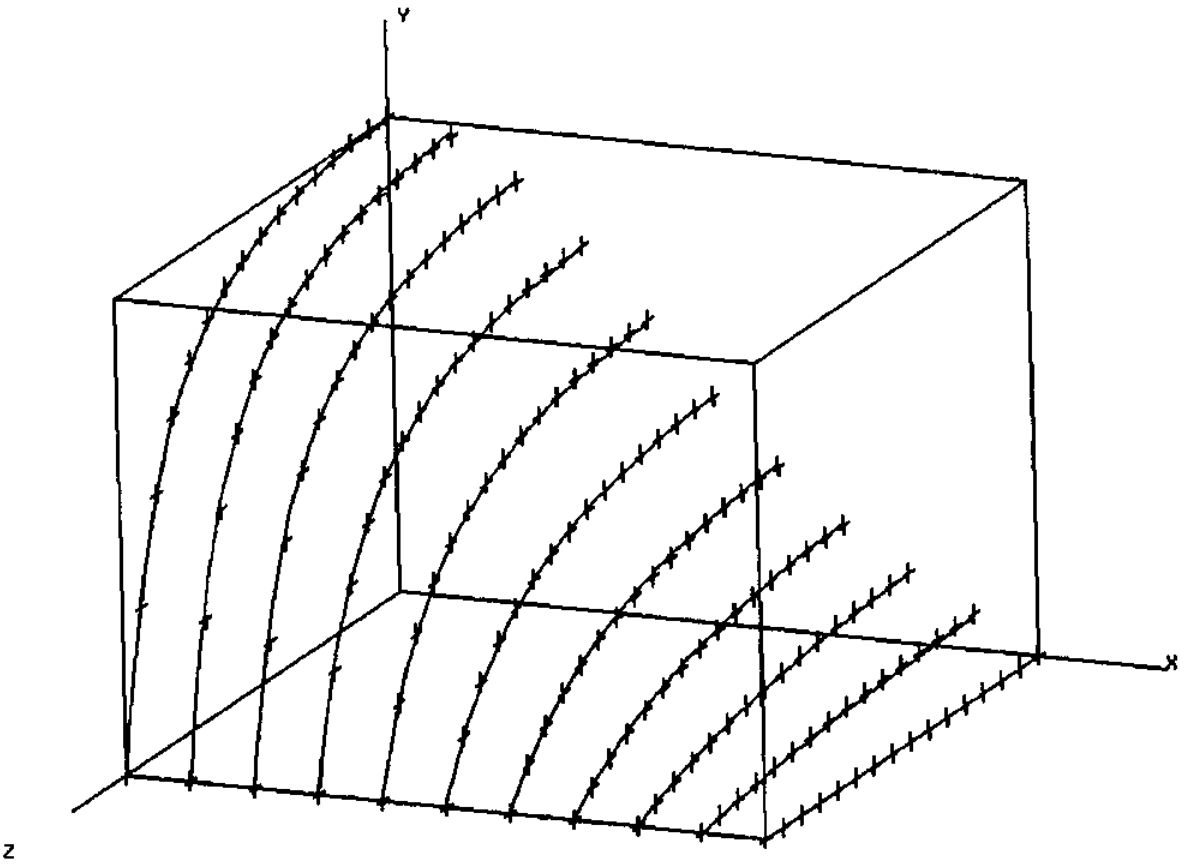
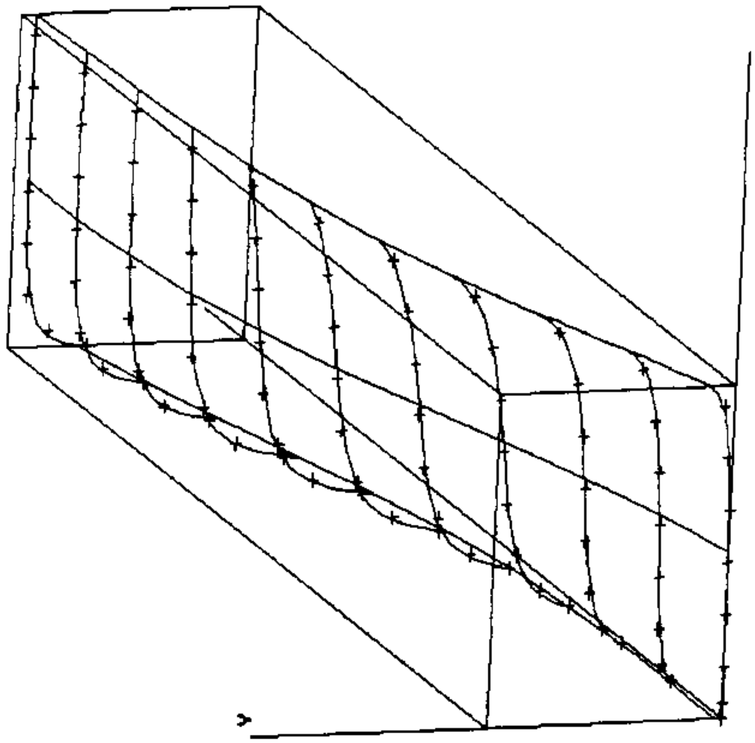


Figure 5-16: Bezier Algebraic Surface Fit of Series 60 Hull Form



presented next. The numerical stability of the singular value decomposition, however, is very good[24].

5.6 Least Squares Solution by Normal Equations Derivation

Reference[23] used the technique of normal equations applied to the homogeneous least squares problem to determine an approximate fit. In this case the normalization used for the distance metric holds the last coefficient in the basis constant.

The first step of this method is the derivation of the normal equations. Given **A**, as above, of size mxn, we compute the Cholesky decomposition of

$$\mathbf{A}^T\mathbf{A} = \mathbf{U}^T\mathbf{U} \tag{43}$$

i.e. we compute the unique nxn upper triangular matrix **U** with non-negative diagonal entries to satisfy equation (43). The least squares problem becomes

$$\mathbf{A}^T\mathbf{A}\ \mathbf{w}= \mathbf{U}^T\mathbf{U}\ \mathbf{w} = 0 \tag{44}$$

This can be simplified further to

$$\mathbf{U}\ \mathbf{w} = 0 \tag{45}$$

The second step of the method differs from the method of normal equations, since we are applying the method to an implicit system in which none of the variables can be identified as independent. In this step the last row of **U** is deleted and we obtain an (n-1)xn upper triangular matrix **U'**. The least squares problem now becomes

$$\mathbf{U'w} = 0$$

and has an exact non-trivial solution. This non-trivial solution can be determined using at most n-1 determinant evaluations[23], provided the original matrix **A** has rank(**A**) $\geq$ n-1. If the rank of matrix **A** is less than n-1, then there is no unique non-trivial solution and the solution fails. If the rank of **A** is n-1, the fit is exact and the residual is $\rho = u_{nn} = 0$.

The quality of the fit (square root of sum of squares of residual vector) is given by u_{nn}, the only non-zero element of the discarded last row of the triangular matrix. More details and a proof of the method, can be found in[23] and[24]. An implementation detail is that instead of the $\mathbf{U}^T\mathbf{U}$ decomposition and in order to improve the stability of the process, the modified Cholesky decomposition was used by us as was also done in[23]. According to this, we decompose

$$\mathbf{A}^T\mathbf{A} = \mathbf{U}^T\mathbf{D}\mathbf{U} \tag{46}$$

where **D** is a diagonal matrix and the leading diagonal of the upper triangular matrix **U** consists of 1's. In this case, again the last row of matrix U is discarded and the modified problem is solved using determinant evaluations[23]. The quality of the fit in this case is $\sqrt{D_{nn}}$ and the last coefficient in the basis is normalized to 1.

The determinant technique described above is equivalent to setting the nth coefficient of **w** to 1, moving the nth column of matrix **A** to the right hand side of equation (34) and then solving for the rest of the coefficients using the ordinary least squares method of normal equations[24].

The determinant technique was also investigated in our work. Each time, the user selects the coefficient of the basis function to be set equal to 1, using information from the geometry of the approximating curve or surface in order to avoid the singularities of such a normalization.

The main advantage of this method is its speed. The time required to form the normal equations is $O(mn^2/2)$, while the Cholesky decomposition and the determinant evaluation require only $O(7n^3/6)$ operations (a total of $O(mn^2/2 + 7n^3/6)$ operations). The main disadvantage of this method, is stability and accuracy[24]. The replacement of **A** by $\mathbf{A}^T\mathbf{A}$ has the destabilizing effect of squaring the condition number of the matrix. When **A** is ill-conditioned, normal equations aggravate the situation. To perform this process, higher precision is required as compared to the singular value decomposition technique.

Reference[23] preferred this technique for efficiency and the capability for fast incremental addition and deletion of points. In order to avoid the numerical problems reference[23] increased the precision used. For the present application, high efficiency was not considered as important. In addition, and more importantly, the technique of normal equations provides a single solution, usually the smallest error solution, and does not provide other possible solutions with equally small residual errors.

5.7 Least Squares Solution by Householder Triangularization

A more accurate alternative to the normal equations solution of the least squares problem is to triangularize matrix **A** directly, using Gauss elimination with partial pivoting or Householder transformations. Using Householder transformations with column pivoting we can decompose the mxn matrix **A** to get

$$\mathbf{Q}\,\mathbf{A}\,\mathbf{P} = \mathbf{U}, \quad \textit{where} \;\; \mathbf{U}^T = [\mathbf{U}'^T, \mathbf{0}] \tag{47}$$

U' is an nxn upper triangular matrix, **0** is an nx(m-n) null matrix, **Q** is the product of the Householder transformation matrices (an orthogonal matrix) and **P** is a column permutation matrix[24]. The least squares solution is obtained in a similar way to the normal equations solution, by approximately solving

$$\mathbf{U}'\,\mathbf{w} = 0 \tag{48}$$

by neglecting the last row of **U'** and solving the modified problem using determinant evaluations to determine the coefficients **w**. The solution of equation (48) is identical to the solution of equation (45).

During this decomposition, pivoting is performed by choosing at each step the column with the largest sum of squares to be reduced next. Thus, this algorithm avoids the singularity problem of setting one of the coefficients of the fit to a constant value. It locates the pivots (diagonal entries of matrix **U'**) in decreasing order (in absolute value). As a result the solution obtained is always the minimum length solution, since the residual of the least squares fit is $|u'_{nn}|$, the last and smallest pivot.

The operation count for this decomposition is $O(mn^2 + 2n^3/3)$, usually larger than the method of normal equations and Cholesky decomposition. It is numerically very stable, but does not provide adequate information about alternative solutions with similarly small residual errors, as is the case with singular value decomposition. Householder triangularization of matrix **A** is usually the first step in the singular value decomposition of matrix **A** to improve its speed and ensure its stability[24].

5.8 Summary

The singular value decomposition technique proved to be the most effective technique for algebraic curve and surface approximation of point data, because of its generality, stability and the insight it provides on the possible set of solutions. Using singular value decomposition, a series of solutions with known and small residuals can be obtained. Some of these solutions can be linearly combined to interpolate points and satisfy other geometric constraints.

There is still no clear automated way of selecting a solution, since additional and unwanted curve or surface segments might appear on the control box, a case which requires user interaction and subdivision. An additional area of investigation, in the approximation of point data using piecewise curves or surfaces (B-spline formulation) is the determination of optimal knot placement to minimize the residual error of the approximation.

6. Interrogation Techniques

This chapter presents some of the interrogation techniques used in the creation and analysis of piecewise continuous algebraic surfaces.

6.1 Contour Display

According to the proposed formulation, as was also indicated in previous chapters, the algebraic surfaces may be easily visualized using planar contours perpendicular to each of the coordinate axes x, y or z. Each of these contours is an algebraic curve defined in a rectangle and can be visualized using techniques for tracing algebraic curves[10 11].

Reliable and efficient tracing of algebraic curves is a topic of active research and development both in academia and industry. Tracing a given algebraic curve in a topologically reliable manner is slower and much more complex than evaluating parametric curves. Tracing methods for algebraic curves can be classified in four main categories[9].

- analytic
- lattice evaluation
- marching
- subdivision

Most of the methods have been developed in the context of interrogation of intersections of polynomial surfaces.

In this work the tracing method developed in our laboratory[11] is used for contouring algebraic surfaces. This tracing method exploits the reformulation of the algebraic equation of the curve in the Bernstein or B-spline basis and uses the geometric properties of the coefficients (weights) as described in Chapters 2 and 3 to obtain a reliable and complete trace of the algebraic curve. Each of the contours is obtained as an intersection of a parametric surface patch and a plane. This method first subdivides the underlying B-spline control surface to its polynomial elements. Then, the significant points of the curve (including border, turning and singular points) are computed using direct numerical techniques, such as minimization and Newton methods, and are used to split the surface into independent subpatches which are processed using adaptive subdivision and faceting techniques. Verification of the computation of all turning and singular points is achieved by interrogation of the derivative patch weights after subdivision at available significant points. In case of repeated failure of the direct numerical techniques to identify turning and singular points, binary subdivision[7] is employed for the computation of such points. This a priori computation of significant points ensures resolution of small details and robustness of the tracing algorithm.

6.2 Ray Tracing Display

An algorithm has been developed in our laboratory for ray tracing of piecewise continuous algebraic surfaces[27]. According to this algorithm, the piecewise algebraic surface is first subdivided in its polynomial Bernstein components using a subdivision algorithm [18] [17]. Each of these components is then ray traced using the following steps.

- Rays from a light source are cast to all the pixels in the computer screen.
- Each of these rays is intersected with the control box containing the algebraic surface patch. Only the rays intersecting the control box need to be further considered and as a result unnecessary computation is avoided.
- If there is an intersection with the control box, the ray line equation is substituted directly into the implicit equation of the surface and the univariate polynomial of the ray intersection with the algebraic surface is obtained. This polynomial has degree, at most, equal to the sum of the degrees of the surface in each of the coordinate directions and is expressed in the Bernstein basis. It is obtained for the different degree algebraic surfaces using a symbolic manipulation system such as Macsyma[28]. The coefficients of this polynomial are symbolic expressions of primitive data.
- Using the variation diminishing property of the Bernstein basis, intervals containing a single root of the polynomial intersection equation are identified using recursive subdivision[29].
- The intervals containing single roots are used in connection with a general nonlinear equation solver[26] to efficiently identify simple roots. The first or all roots within the box of interest may be identified in this manner.
- At each of the intersection points the normal to the algebraic surface is obtained and an appropriate lighting model (Phong model[30]) is employed to determine the intensity of the corresponding pixel in the computer screen.

Figures 6-1 to 6-7 display some ray traced images of the algebraic surfaces presented in the previous chapters. Figure 6-1 displays the first octant of a sphere, while Figures 6-2 and 6-3 present the "pulled" and "pushed" octant of the sphere as were these also presented in Figures 4-4 and 4-5. Figure 6-4 presents the ellipsoid, while Figure 6-5 presents the doubly curved blending type of surface which was obtained in the previous Chapter using least squares fitting of points. Figures 6-6 and 6-7 present the Wigley Hull and the Series 60 type of ship hull approximated in Chapter 5 and also shown in Figures 5-14 and 5-16. Implementation of the above algorithm was performed on a DEC Vaxstation II driving a Silicon Graphics 3030 workstation through the IRIS remote graphics library to display the image. The time required for each of these images depends on the order of the algebraic surface and on the size of the control box with respect to the graphics screen.

6.3 Volume and Moment of Inertia Evaluation

The volume enclosed on one side of an algebraic surface and the control box as well as the various moments of inertia of the enclosed volume about the x, y or z coordinate axes were obtained using an octree subdivision technique. The algorithm used exploits some of the geometric properties of the coefficients (weights) of the algebraic surface representation.

A point $P = (x_i,y_i,z_i)$ is considered to be inside or on an implicit algebraic surface provided $f(x_i,y_i,z_i) \leq 0$, while it is considered to be in the exterior of the algebraic surface if $f(x_i,y_i,z_i) > 0$. The important property of the present formulation to use in the octree algorithm is the following. If the coefficients (weights) of a control box are all non-positive, then $f(x,y,z) \leq 0$ and the whole box is contained within a surface. Alternatively, if the coefficients (weights) of a control box are all positive, then $f(x,y,z) > 0$ and the control box is exterior to the surface. This is a direct result of the non-negativity of each of the B-spline basis functions.

According to the octree algorithm implemented, the control box of the piecewise algebraic surface is subdivided using the Oslo algorithm[17] into eight smaller subboxes. Each of these subboxes is checked for the following:

- If all the coefficients of the subbox are positive, there is no part of the surface inside this subbox and the subbox can be neglected.
- If all the coefficients of the subbox are non-positive, the subbox is contained within the surface and its volume and moments of inertia about the coordinate axes can be added to the total volume and moments of inertia enclosed by the surface.
- If there are coefficients with both signs, then the subbox contains a portion of the surface and further subdivision is required.

The subdivision is terminated, when the dimensions of the subboxes reach a predefined limit (e.g. 0.1% of the original size of the control box). At the termination point some of the subboxes are still undecided since these subboxes contain a small portion of the surface. For

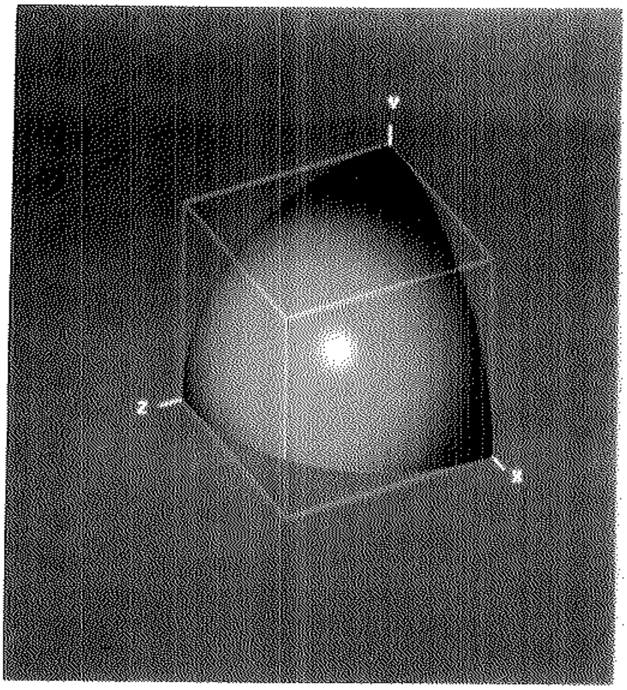


Figure 6-1: Ray Traced Image of an Octant of a Sphere

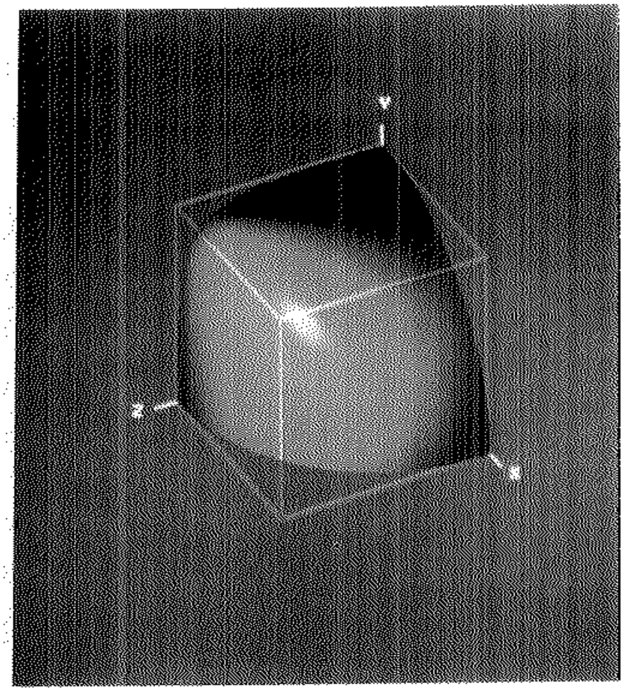


Figure 6-2: Ray Traced Image of a "Pulled" Octant of a Sphere

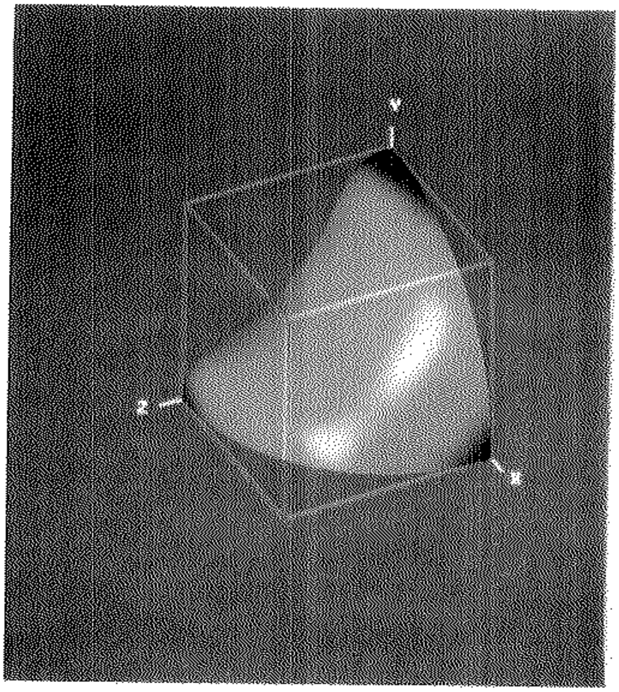


Figure 6-3: Ray Traced Image of a "Pushed" Octant of a Sphere

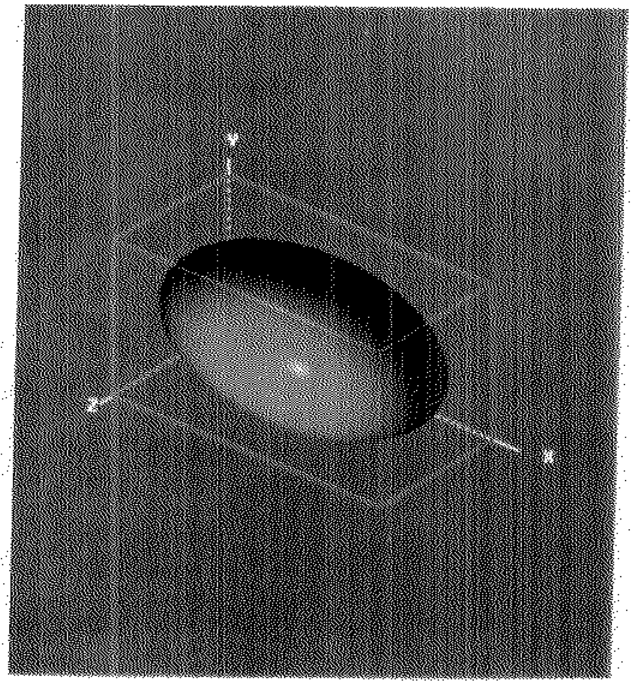


Figure 6-4: Ray Traced Image of an Ellipsoid

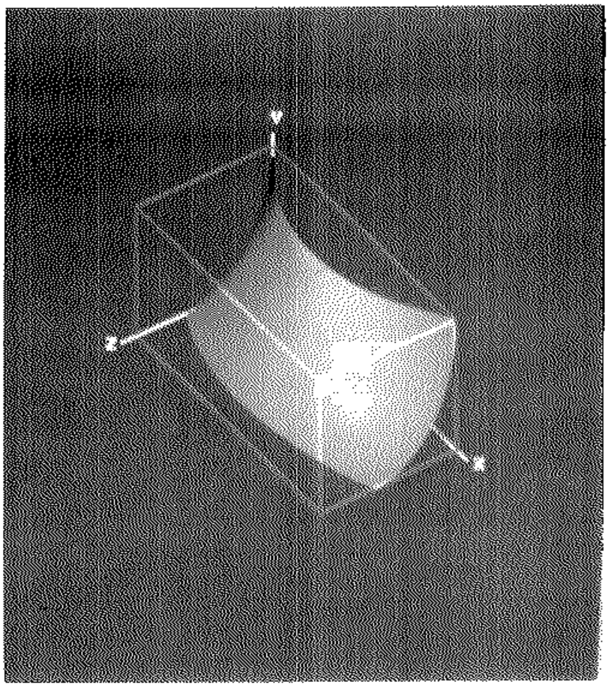


Figure 6-5: Ray Traced Image of a Type of Blending Surface

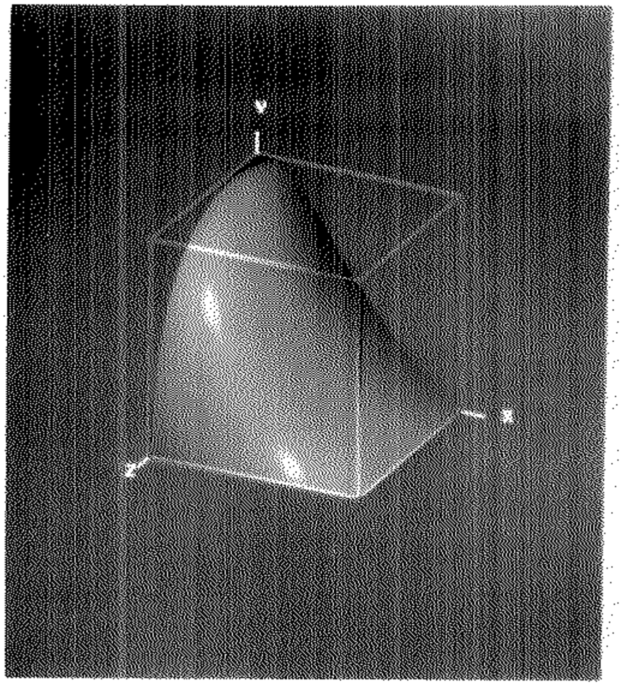


Figure 6-6: Ray Traced Image of the Wigley Hull

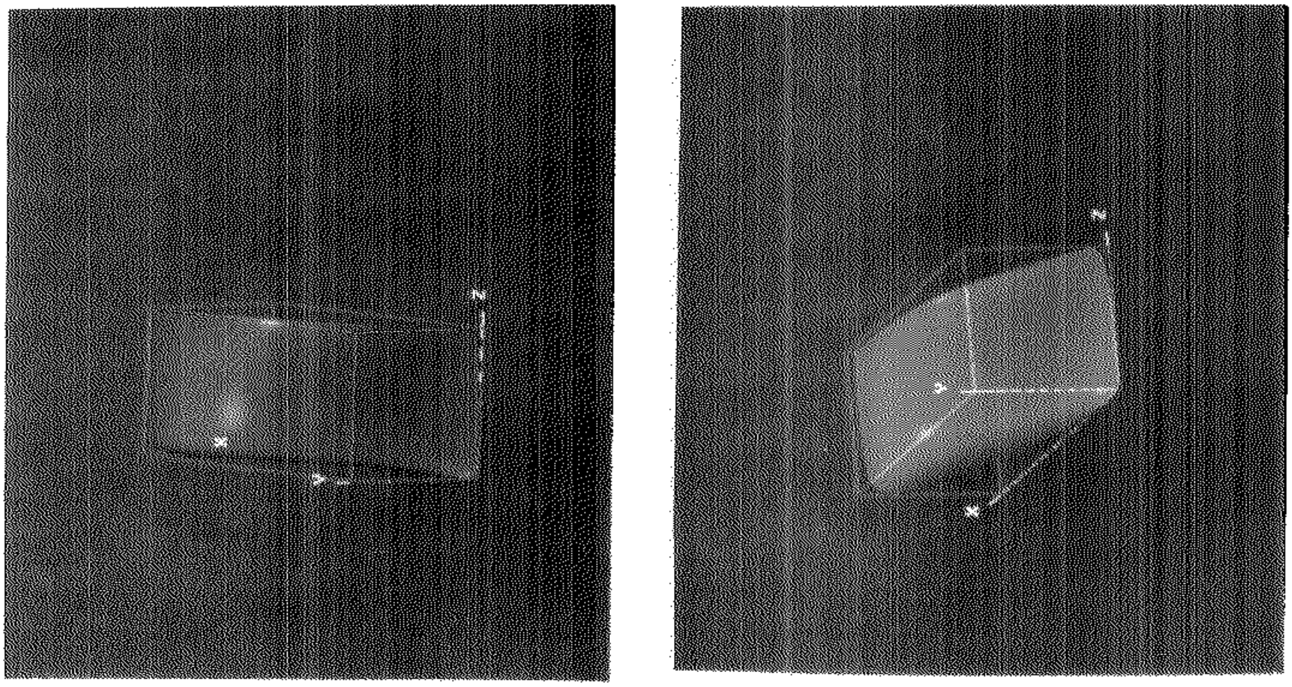


Figure 6-7: Ray Traced Image of the Series 60 Type Hull

the final volume and moment of inertia estimates we can use only the subboxes interior to the surface to obtain a lower bound to the volume and inertia estimates, or we can use both the interior and undecided subboxes to obtain an upper bound to these estimates. A more accurate approximation will be to use the interior subboxes and half of the contribution from the undecided subboxes for faster convergence to the actual volume and moment of inertias enclosed by the algebraic surface.

The octree algorithm implemented is very efficient and does not require detailed tracing of the algebraic surface. Table 6-1 presents an example showing the convergence of the volume enclosed by the first octant of a unit sphere in a unit rectangular box.

Table 6-1: Convergence of Volume Estimate Enclosed by the First Octant of a Unit Sphere. Exact Volume, $\pi/6 = 0.523599$

Minimum Subbox Size	Undecided Subboxes	Lower Bound	Upper Bound	Volume Estimate
0.5	7	0.4670	0.5625	0.514765
0.1	34	0.5050	0.5313	0.518150
0.01	145	0.5176	0.5244	0.521006
0.001	595	0.5221	0.5238	0.522941
0.0001	2404	0.5232	0.5236	0.52340

6.4 Curvature Evaluation

An important tool in the design of free form shapes using the algebraic surface formulation is the capability of evaluating the curvature variation on the surface to verify the absence of undesirable features and oscillations of the surface.

In the present implementation, we have developed a capability to evaluate and display the curvature of the surface on the planes containing the contours used to display the surface. Since each of the surface contours is obtained as the trace of an algebraic curve in a plane, the curvature of this contour can be obtained similarly as the curvature of the corresponding planar algebraic curve. The curvature k of an implicit curve defined by f(x,y) = 0 is given by

$$k = -\frac{f_{xx}f_y^2 - 2f_{xy}f_xf_y + f_{yy}f_x^2}{(f_x^2 + f_y^2)^{3/2}} \tag{49}$$

An example illustrating the use of curvature is presented in Figures 6-8 and 6-9. These figures display the curvature on two of the algebraic curve solutions of the fitting example presented in the previous Chapter. The curvature lines of the algebraic curve and of the parametric B-spline curve used to obtain the fit are shown. In these Figures, one can see the degree to which the algebraic curve curvature approximates the curvature of the initial parametric B-spline curve which has a discontinuity also at the internal knot and identify the existence of undesirable curvature fluctuations or additional inflection points.

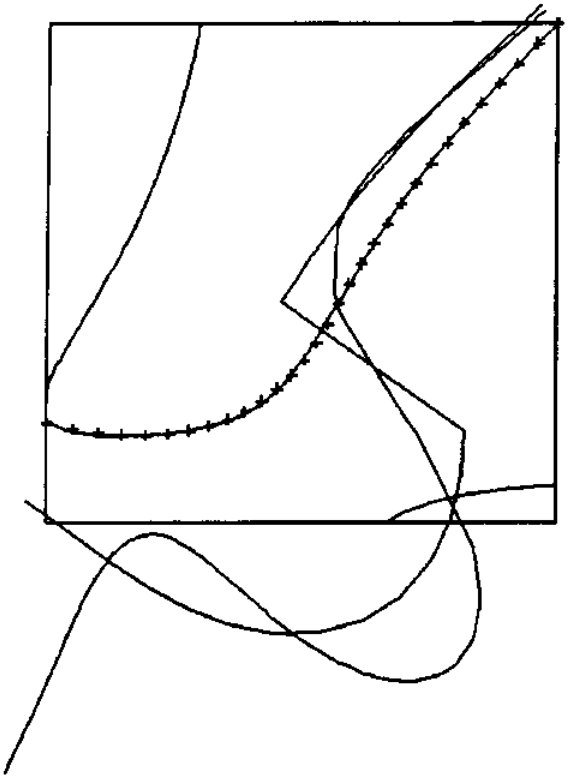


Figure 6-8: Curvature Lines for Smallest Singular Value Algebraic Curve Fitted Using B-spline Curve Parametric Points

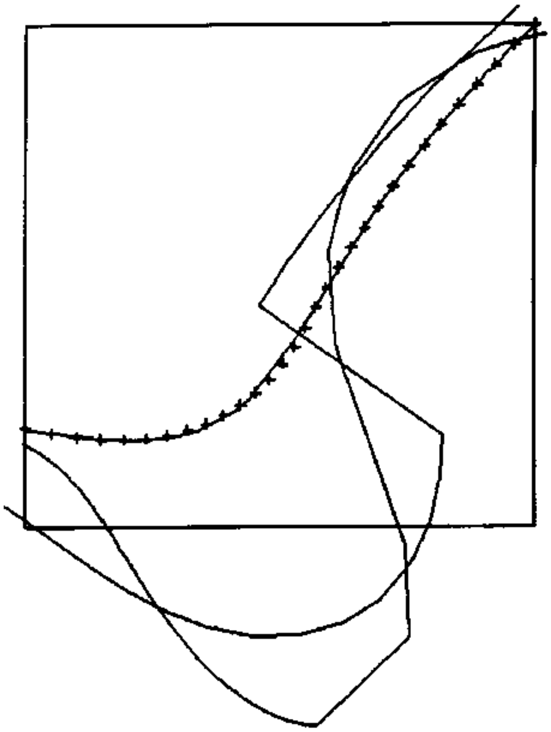


Figure 6-9: Curvature Lines for Better Algebraic Curve Fitted Using B-spline Curve Parametric Points

7. Conclusions and Recommendations

A method of representing algebraic surface patches within a rectangular box was described and extended to handle piecewise continuous algebraic surface patches. Techniques to control and manipulate the shape of the algebraic surface patch within a rectangular box were also presented together with some useful properties of this representation. The creation/approximation of lower dimensional data with piecewise continuous algebraic surface patches using least squares was also described in detail. Several interrogation techniques useful in the analysis of piecewise continuous algebraic surfaces were also presented.

As was seen in the previous chapters, low order piecewise continuous algebraic curves and surfaces allow representation of a large class of fairly complex sculptured shapes and therefore provide a viable alternative to modeling shape with piecewise polynomial parametric surface patches. Modeling with low order algebraic curves and surfaces in rectangular boxes in terms of the B-spline basis provides a capability for piecewise continuous shape representation, allowing for continuity or discontinuity control of curves and surfaces and the representation of complex artifacts. The geometrical significance of the algebraic curve and surface coefficients facilitate intuitive understanding, interactive refinement and reliable and efficient processing of the representation.

The degree reduction in the resulting representation, facilitates computationally complex interrogation problems. Low order algebraic surfaces lead to intersection curves of lesser degree compared to the intersections of the widely used rational polynomial biquadratic and bicubic patches. In the algebraic patch case within rectangular boxes defined by triple B-spline products, intersections of two patches of degree two in each variable result in an algebraic curve of degree eight in each variable. Similarly, for patches of degree three in each variable, the intersection curve is of degree eighteen. Similar intersections for biquadratic and bicubic rational patches result in algebraic curves with degrees sixteen and fifty four in each variable, respectively. Similarly low order algebraics lead to low degree algebraic blending surfaces[2].

Low order algebraic surfaces naturally permit the creation of a class of piecewise continuous free-form volume primitives within rectangular boxes to be used in a

Constructive Solid Geometry environment. Closed volumes, as well as portions of closed volumes can be modeled in a unified environment (cubes, ellipsoids, general surfaces with the same representation) allowing efficient and robust implementation of Boolean combinations of volumes, interference preprocessing algorithms etc.

Another application of the least squares analysis techniques presented in this work is in the field of Computer Vision. The majority of the work on least squares fitting of algebraic curves and surfaces comes from this area, arising from creation of curve and surface representations from measured point data.

Representing shape with algebraic surfaces is not as intuitive as representing them with parametric polynomial surfaces. In addition, there are limitations to the shape complexity which can be represented by low order algebraic curves and surfaces of the type studied here and this is directly related to the reduced flexibility of the representation such as the inability to directly provide three-dimensional curve boundaries on the patches and to control the slopes across such boundaries. In the modeling system studied here, such boundaries may be approximately constructed using primitive intersections. The degree of flexibility of such surfaces may be enhanced in specialized applications by changing the shape of the control box. For example, extensions of the proposed methodology to representations within non-rectangular but hexahedral boxes such as cylindrical or spheroidal may provide additional advantages for specialized sculptured objects constructed in terms of cylindrical or spherical sections.

An area of future attention in shape creation is the use of control curves obtained from least squares approximation of points to generate the three-dimensional shape in a sweep-type operation. The problem of fairing piecewise continuous algebraic curves and surfaces needs also to be addressed to avoid unwanted curvature fluctuations or large higher derivative discontinuities on the curve or surface representation.

Another area of future attention is the use of the proposed formulation to represent shape subject to constraints such as tangencies in faces, flat portions e.t.c.. Sometimes it is desired to generate surfaces with specific normal vectors at one of the faces of the bounding box (e.g. surface tangent to a face along a particular curve on that face). In order to achieve this, a higher order reducible algebraic equation with multiple components is required. It is worthwhile to examine the use of piecewise continuous algebraic B-spline surfaces of lower degree to approximate the high order algebraic equation in such regions.

The development of a modeling system with piecewise low order algebraics, also requires research on reliable and efficient interrogation operators, such as intersection, offset and blend. Intersection operators are, for example, needed in modeling the internal subdivision and structural reinforcement of vehicles with free-form bounding surfaces such as almost all marine and aerospace structures, and in numerous analysis and fabrication tasks. Offset operators are required in numerical control manufacturing, representation of material shells, such as the plating of ships, airplanes and automobiles, dimensional tolerance and access space representation for manufacturing simulation and robotic applications. In addition, intersections and offsets are needed in shape feature recognition for automated analysis. Finally, blends are needed in the representation of smooth transitions between surfaces either to provide explicit specification of manufacturing operations or to meet important functional requirements, such as structural and hydrodynamic.

References

1. Sederberg, T. W., "Planar Piecewise Algebraic Curves", *Computer Aided Geometric Design*, Vol. 1, 1984, pp. 241-255.

2. Hoffmann C., Hopcroft J., "The Potential Method for Blending Surfaces and Corners", *Geometric Modeling: Algorithms and New Trends*, SIAM, 1987, pp. 347-365.

3. Farouki, R. T., Hinds, J. K., "A Hierarchy of Geometric Forms", *IEEE Computer Graphics and Applications*, Vol. 5, No. 5, May 1985, pp. 51-78.

4. Farouki, R. T., "The Characterization of Parametric Surface Sections", *Computer Vision, Graphics and Image Processing*, Vol. 33, 1986, pp. 209-236.

5. Sederberg, T. W., Anderson, D. C., Goldman, R. N., "Implicit Representation of Parametric Curves and Surfaces", *Computer Vision, Graphics and Image Processing*, Vol. 28, No. 1, 1984, pp. 72-84.

6. Sederberg, T. W., *Implicit and Parametric Curves and Surfaces for Computer Aided Geometric Design*, PhD dissertation, Purdue University, August 1983.

7. Geisow, A., *Surface Interrogations*, PhD dissertation, School of Computing Studies and Accountancy, University of East Anglia, Norwich NR47TJ, U. K., July 1983.

8. Sederberg, T. W., "Piecewise Algebraic Surface Patches", *Computer Aided Geometric Design*, Vol. 2, 1985, pp. 53-59.

9. Patrikalakis, N. M., "Piecewise Continuous Algebraic Curves in Terms of B-Splines", *Submitted for Publication*, February 1987.

10. Patrikalakis, N. M., Prakash, P. V., *Computation of Algebraic and Polynomial Parametric Surface Intersections*, MIT Sea Grant Report No. 87-19 , 1987.

11. Prakash, P. V., *Computation of Surface-Surface Intersections for Geometric Modeling*, PhD dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1988.

12. Kahan, W., "A Survey of Error Analysis", *Proceedings of the International Federation for Information Processing Congress 1971*, Vol. 2, August 1971, pp. 1214-1239.

13. Farouki, R. T., Rajan, V. T., "On the Numerical Condition of Bernstein Polynomials", *IBM Research Report RC 12626*, March 1987.

14. Sederberg, T. W., Parry, S. R., "Comparison of Three Curve Intersection Algorithms", *Computer Aided Design*, Vol. 18, No. 1, January 1986, pp. 58-63.

15. De Boor, C., "On Calculating with B-Splines", *Journal of Approximation Theory*, Vol. 6, 1972, pp. 50-62.

16. Gordon, W. J., Riesenfeld, R. F., *B-Spline Curves and Surfaces*, Computer Aided Geometric Design, Edited by Barnhill, R., and Riesenfeld, R. F., Academic Press, Inc., 1974.

17. Cohen, E., Lyche, T., Riesenfeld, R., "Discrete B-Splines and Subdivision

Techniques in Computer-Aided Geometric Design and Computer Graphics'', *Computer Graphics and Image Processing*, Vol. 14, 1980, pp. 87-111.

18. Lyche, T., Cohen, E., Morken, K., ''Knot Line Refinement Algorithms for Tensor Product B-Spline Surfaces'', *Computer Aided Geometric Design*, Vol. 2, 1985, pp. 133-139.

19. Chryssostomidis, C., ''Computer Aided Interference Control and Space Reservation'', *Proceedings of the Conference on Shipyard Operation and Ship Design, Tokyo*, 1973.

20. Cohen E., Lyche T., Schumaker L., ''Algorithms for Degree Raising of Splines'', *ACM Transactions on Graphics*, Vol. 4, No. 3, 1985, pp. 171-181.

21. Carlson R. H., ''Design of an Algebraic Surface Editor''. Naval Engineer's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1988.

22. Boehm W., ''Inserting New Knots Into B-Spline Curves'', *Computer Aided Design*, No. 12, 1980, pp. 99-102.

23. Pratt, V., ''Direct Least-Squares Fitting of Algebraic Surfaces'', *ACM Computer Graphics*, Vol. 21, No. 4, 1987, pp. 145-152.

24. Lawson, C. L., Hanson, R. J., *Solving Least Squares Problems*, Prentice Hall Inc, Englewood Cliffs, NJ., 1974.

25. Abbott I. H., Von Doenhoff A. E., *Theory of Wing Sections*, Dover Publications, Inc., New York, 1959.

26. NAG, *Numerical Algorithms Group FORTRAN Library*, NAG, Oxford, England, 1985.

27. Moran B. A., ''Ray Tracing Piecewise Polynomial Surface Patches'', Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1988 (in preparation).

28. Symbolics Inc., *VAX UNIX MACSYMA Reference Manual*, Symbolics Inc., 1985.

29. Lane, J. M., Riesenfeld, R. F., ''Bounds on a polynomial'', *BIT: Nordisk Tidskrift for Informations-Behandling*, Vol. 21,, No. 1,, 1981, pp. 112-117.

30. Foley J. D., Van Dam A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Company, 1982.